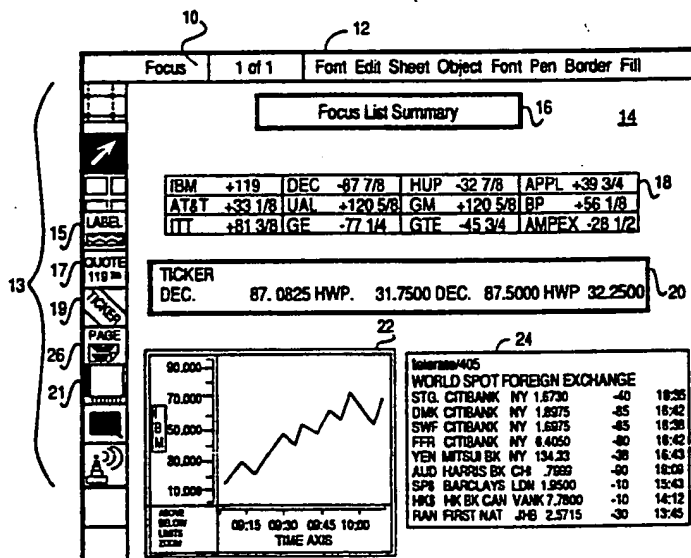




## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification <sup>5</sup> :  G06F 15/20	A1	(11) International Publication Number: WO 92/12488  (43) International Publication Date: 23 July 1992 (23.07.92)
(21) International Application Number: PCT/US91/09811 (22) International Filing Date: 20 December 1991 (20.12.91) (30) Priority data: 636,044                      28 December 1990 (28.12.90) US (71) Applicant: TEKNEKRON SOFTWARE SYSTEMS, INC. [US/US]; 530 Lytton Avenue, Palo Alto, CA 94301 (US). (72) Inventors: RISBERG, Jeffrey, Scott ; 3249 Morris Drive, Palo Alto, CA 94303 (US). SKEEN, Marion, Dale ; 3826 Magnolia Drive, Palo Alto, CA 94306 (US). (74) Agents: FISH, Ronald, C. et al.; Skjervén, Morrill, Mac- Pherson, Franklin & Friel, 25 Metro Drive, Suite 700, San Jose, CA 95110 (US).		(81) Designated States: AT (European patent), AU, BE (Euro- pean patent), CA, CH (European patent), DE (Euro- pean patent), DK (European patent), ES (European pa- tent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, KR, LU (European patent), MC (European patent), NL (Euro- pean patent), SE (European patent).  Published With international search report.

(54) Title: APPARATUS AND METHOD FOR CREATION OF A USER DEFINABLE VIDEO DISPLAYED DOCUMENT SHOWING CHANGES IN REAL TIME DATA



## (57) Abstract

A computer system (92) for providing a facility wherein a "living document" (66) comprised of multiple pages (26), each of which contains any one of a number of different items of information (18) some of which may reflect the current value of real time data and reflect changes in the value thereof in real time. The items of information on each page are defined as to content by the user and the layout of each page in terms of position and style such as font, color, presentation type, e.g., graph (21), ticker etc., is defined by the user. The user may also set alarm limits (28) for any real time value and define scripts of commands which are to be executed in case an alarm limit (28) is exceeded. The user may also define "buttons" (19) which may be activated graphically (21) or otherwise by the user. Associated with each button is a script of one or more commands which are executed when the button is activated. The system is designed to work in conjunction with the Teknekron Information Bus which is the subject of EPO publication 412, 232, A2, published February 13, 1991, to obtain real time data for display (14) on the "living document" (66).

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	ES	Spain	MG	Madagascar
AU	Australia	FI	Finland	ML	Mali
BB	Barbados	FR	France	MN	Mongolia
BE	Belgium	GA	Gabon	MR	Mauritania
BF	Burkina Faso	GB	United Kingdom	MW	Malawi
BG	Bulgaria	GN	Guinea	NL	Netherlands
BJ	Benin	GR	Greece	NO	Norway
BR	Brazil	HU	Hungary	PL	Poland
CA	Canada	IT	Italy	RO	Romania
CF	Central African Republic	JP	Japan	RU	Russian Federation
CC	Congo	KP	Democratic People's Republic of Korea	SD	Sudan
CH	Switzerland	KR	Republic of Korea	SE	Sweden
CI	Côte d'Ivoire	LI	Liechtenstein	SN	Senegal
CM	Cameroon	LK	Sri Lanka	SU	Soviet Union
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
DE	Germany	MC	Monaco	TG	Togo
DK	Denmark			US	United States of America

- 1 -

Specification  
APPARATUS AND METHOD FOR CREATION OF A USER  
DEFINABLE VIDEO DISPLAYED DOCUMENT SHOWING  
CHANGES IN REAL TIME DATA

5 BACKGROUND OF THE INVENTION

The invention pertains to the field of application programs for monitoring and managing complex systems with many variables having values which vary in real time. More specifically, the invention pertains to the field of  
10 software systems which permit the user to create custom active documents with the layout, look and content defined by the user and displayed on a video display. Changes in the values of real time data which are included in the active document defined by the user are reflected  
15 immediately on the display.

In the management of complex systems such as the financial community, airplanes, semiconductor fabrication processes, etc. it is often useful for a user such as a financial trader to be able to look at only some subset of  
20 the total data available to him or her or to arrange the available data in a style which best suits the user's management and/or analysis style. In the prior art for the financial community, the Quotron product fills some portion of this need, but is inadequate in many respects. The  
25 Quotron product has a video display where three basic areas are available for customization by the user. One area is reserved for stock quotes where individual stock prices are displayed in a block. The user can customize to the extent of defining which of the many stocks for which quotes are  
30 desired. The current price of the stock is then displayed in each block or window devoted to that stock by network access of a service that provides stock quotes. Changes in the stock price are reflected on the display when they occur. Another area of the display is reserved for so-  
35 called "tickers", i.e., streams of trade data for various

- 2 -

stocks defined by the user with the stream for all the trade data moving across a common window. A third area of the display is reserved for a market monitor display where a plurality of stock prices are displayed as a group in a single window. With the content of the group, i.e., the stock prices selected for display selected by the user.

The Quotron product is inadequate in many respects. For example, it is useful for a user to be able to be able to define a multi-sheet, completely custom document with whatever data he or she desires on each sheet arranged in whatever groupings, locations, format, color, font, pen desired with changes in real time data being reflected immediately on the display. Further, it is useful for the user to be able to set upper and/or lower alarm limits for any or all real time data displayed on any sheet and to define scripts of actions to take if the value of the real time data exceeds these limits. Further, it is useful to be able to publish any or all the data shown on any sheet of the active document on a network to which other users and/or automated processes are coupled via their respective hosts. Finally, it is useful to be able for a user to define and layout certain areas which will appear on all sheets such as one to four margins, headers or footers with text, a common "button" or a set of buttons, i.e., display objects with user programmable text labels and a "push" operation with a programmable action which occurs every time upon a "push" of the button such as by selection of the button with a mouse, touch screen, voice designation or by the keyboard etc.

#### SUMMARY OF THE INVENTION

30 An apparatus and method according to the teachings of the invention provides a computer facility (hereafter the application or program) whereby a user, using a collection of layout tools may define an active document. "Active document" as that term is used herein means a video  
35 displayed document of one or more "sheets" of the user's design which incorporates text, displays of real time data

- 3 -

in user definable style, e.g., color, font, background, pen size etc. and format, e.g., quote, ticker, graph etc., alarms, and alarm scripts, i.e., user defined scripts of commands to be processed (much like a word processing or 5 spreadsheet macro) when an alarm limit is exceeded. The program automatically accesses the network to which the host is connected through network interface software which establishes the proper subscriptions for the desired real time data with the appropriate information service and the 10 server upon which this service process is running. Real time data is then passed to the program from whatever network communication process is being used and is immediately displayed in the format, style and location previously specified by the user. Although the invention 15 will hereafter be described in the preferred embodiment for use in a financial environment such as a trading floor of a broker such as Dean Witter etc., the invention is not limited to such applications. Any complex system which generates real time data which control operators must 20 monitor are subject to being monitored and controlled using the teachings of the invention.

The user defines each sheet of the document using a variety of tools which can be used to create display objects such as simple quotes or tickers to sophisticated graphs and 25 tables. The user may also use the tools to create text fields such as customized help screens which help other users understand the active document which has been created. All the tools share a common set of commands which are located in an menu typically displayed at the top of each 30 sheet. Preferably, the tools are symbolized by icons and the menu options, when invoked, cause pop-up menus to appear with other options. The menu options are used to set the symbols for stock prices to be monitored, changing display characteristics and selecting which of the multiple pages to 35 view. The location and type of menu display and the location and type of system used to display and/or invoke the layout tools is not critical to the invention.

- 4 -

Similarly, the type of input device used to select tools, specify locations, specify symbols or the source of real time data to display at any particular display object, or to direct the program where on a sheet to display a particular display object is not critical to the invention. A mouse, trackball, digitizer, keyboard, voice processor and map coordinate system, touchscreen, or any other present or future device may be used such as a thought processor.

The tools provide access to information from any source including other programs running on the same host or somewhere else on the network, ticker plants, information services or databases. In the preferred embodiment, the program can support data feeds from Reuters Market Feed 2000/IDN, Telekurs Ticker, CMQ Telerate MarketFeed, Canquote, and Quotron. In addition, the program (known commercially as the MarketSheet™ facility or program) can accept prices extracted from paged market data feeds such as Telerate TDPF, Reuters RDCDF, FBI and RMJ. These types of data are first extracted by Page Shredder, another program available from the assignee of the present invention.

In the preferred embodiment, the tools available for defining an active document are as follows. A label tool allows the user to enter static text to label or annotate the active document or to create his or her own personalized help screens. A quote tools displays the value of an issue, including a user defined set of other fields pertaining to that particular company in a display style specified by the user. For example, a brief style displays only the price where a comprehensive style displays all the available fields. A ticker tool can be used as a selective or block ticker, and can show data in any display style. Upticks and Downticks can be shown in color and volume information can be included. A page fragment tool displays a region of a page-based feed such as Telerate or Reuters. Any region of the page designated by the user can be displayed from a single character to a full page. Highlighting modes are

- 5 -

provided to highlight that has changed. A time based graph tool can be ;ed to create graph display objects to graphically display the changes in value of a variable such as price per time. The time and price axes may be scaled to 5 minutes or seconds, and the price value may be set to any unit such as 1/8 or 1/32 of a dollar. Above and below channel segments can be drawn on the graph. Graphs may be merged to show two issues against the same time axis. A data set graph tool can be used to create graph display 10 objects which display the values of multiple instruments such as stocks or bonds or other subscribed values in real me such as a yield curve in a semiconductor processing application environment. Graphs may be merged to show two different sets of issues against each other to indicate 15 market opportunities. A table tool can be used to create display objects which show position blotters, currency lookup tables, and names of commonly used pages securities. A publisher tool publishes information constructed using the program of the invention or entered the user onto the 20 network using the network communication process running in the environment in which the program or process of the invention is running. The published information can be used by other processes linked to the network or as a bulletin board for use by her traders. A button tool can be used to 25 create splay objects that execute scripted actions when the button is "pushed", i.e., selected in any way on the splay such as by clicking on the button by a mouse. The scripted actions are entered by the user in whatever sequence is desired in a language such as the MarketScript™ command 30 language comprised of all commands at the program according to the teachings of the invention can execute. In the preferred embodiment, the scripted command sequence can also include commands to the operating system, the network communication software and other processes running on the 35 same host or elsewhere on network. Buttons can be programmed to carry out commonly performed operations such as moving quickly to an important page or performing an

- 6 -

operation to be carried out when an alert condition occurs. The buttons allow creation of hypertext links between different sheets, and the alert scripts can perform operations such as changing a color, flashing an object, 5 sounding an audible alarm or executing an external program. The latter capability provides great flexibility by enabling clients to program features such as the ability to telephone a beeper service when an alert occurs. A "glossary" facility allows users to add new operations, i.e., commands, 10 to the scripting language and customize the menus as needed.

For ease of learning the system, new users are supported by providing short and full menu sets. New users can use the short set while working with existing templates and tutorial files. After the system has been learned, the 15 user can graduate to the full menu set.

In the preferred embodiment, the program is used in a distributed system architecture embodied in a local area network coupling workstations and personal computers such that the right amount of processing power can be dedicated 20 to each user and application. The distributed system architecture also allows critical applications to be supported by redundant or complementary equipment such that the system is sufficiently reliable to continue to operate even if a market data feed or server should fail.

25 In the preferred embodiment, the program according to the teachings of the invention is ported to work with the TIB™ suite of network communication programs available from the Teknekron Software Systems in Palo Alto, California. The TIB software is described in U.S. patent applications 30 having serial numbers 386,584 filed July 27, 1989; 07/601,117 filed October 22, 1990; and 07/632,535 filed December 21, 1990, which are hereby incorporated by reference. The TIB software supports subject based addressing, network architecture decoupling, communication 35 protocol decoupling, data decoupling and separation of information sources from consumers. The TIB software subject based addressing capability frees applications such



- 7 -

as the program according to the teachings of the invention from the need to know exactly where on the network information of a particular type can be obtained. For example, a trader can reference "long bond yield" as the subject of a desired real time data subscription without the need to know the specific physical location on a specific page of a specific information service running on a specific server. Then, if the source of the information changes, users need not change their screens and programmers need not change their programs.

The separation of information sources and consumers allows users to add or change market data sources without changing the application that use the data. The users can also add and change applications without impact on the existing sources of information. For example, data values from fixed format pages of market data can be combined with data from elementized feeds in a realtime spreadsheet. The spreadsheet can compute a result, e.g., the theoretical value of a derivative security, and publish it on the network through the TIB software. The program of the invention can then graph the value in real time. None of these application programs need ever be aware of the existence or operation of the others. New application programs can be added to a firm's computing repertoire at any time, building on each other to create a total information environment.

The menu of commands allows the user to display an index of the sheets which have been defined for a particular active document file, and to select the sheet to view. The menu options also include commands to manage sheets and sheet files, and to control the appearance of the display and the objects within it. The menu also includes commands to rearrange the location of the display object windows or boxes (the term boxes will be used herein to avoid confusion with the term windows in which separate processes may be running in multitasking environment or DOS windows environments). In the preferred embodiment, the display

- 8 -

object boxes may be layered on top of each other, and the menu includes options to change the order of the layers and move any particular box to the top of a stack. In the preferred embodiment, only one sheet is shown on the display  
5 at any particular time, but in alternative embodiments, several sheets may be shown in separate "windows" or layers on the display.

BRIEF DESCRIPTION OF THE DRAWINGS:

Figure 1 is a typical sheet layout in an active  
10 document containing one or more user defined sheets constructed with a program according to the teachings of the invention.

Figure 2 is a state diagram for the normal and alert states to illustrate the operation of alert scripts.

15 Figure 3 is a diagram of a typical environment in which the program of the invention operates.

Figure 4 is an example of an instance of an Active Object class.

Figure 5 illustrates how the properties of an Active  
20 Object are stored as data structures in memory, and how libraries of programs are linked to these data structures to carry out the operations associated with each instance of an Active Object.

Figure 6 is a block diagram of the parts of a program  
25 according to the teachings of the invention, and the components of the computing environment in which the program, according to the teachings of the invention, operates.

Figure 7 is a block diagram of the parts of an Active  
30 Object.

Figure 8 is a flow diagram of processing after a real time data update.

Figure 9 is a diagram of event processing flow in a program according to the teachings of the invention.

35 Figure 10 is a drawing illustrating the basic display of a program according to the teachings of the invention.

- 9 -

Figure 11 is an illustration of the menu bar.

Figure 12 is a dialog box example.

Figure 13 is an example of the pull-down sheet menu.

Figure 14 is an example of the dialog box to select the 5 index of interest.

Figure 15 is the tools icon menu.

Figure 16 is an example of how the display for an Active Object changes when the object has been selected.

Figure 17 is an example of the display for multiple 10 Active Objects when they have been selected.

Figure 18 is an example of an object which has been selected by keyboard transversal.

Figure 19 is an example of overlapped Active Objects.

Figure 20 shows the pull-down menu structure for all 15 menu options on the menu bar.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT:

Referring to Figure 1, there is shown one example of a user defined sheet in an active document constructed using a program according to the teachings of the invention. At 20 the top left corner of the sheet is the file name, in this case FOCUS, naming the file in which the active document of which the user programmed sheet of Figure 1 is a part. A menu of commands, i.e., functions which can be invoked by the user through whatever input device is being used, is 25 shown at 12. In the preferred embodiment, the menu choices shown at 12 each "overlie" a "pop-up" menu displaying further command choices related to the overlying command choice. Those pop-up menus will be detailed in a later figure. The menu lets the user display an index of sheets 30 within the active document and to select the sheet to view. The menu 12 also includes options to manage sheet programming and active document files. The menu 12 also includes commands to control the appearance of the display and the display objects associated with the Active Objects 35 programmed by the user into a sheet.

Icons representing the tools that are available to the

- 10 -

user to define the Active Objects which are to appear on any particular sheet of an active document are shown at 13. Each tool, when invoked, allows the user to create an Active Object on the programmable portion of the display at a location specified by the user using some input device. Each tool can be used to define an instance of a particular class of Active Objects or to carry out an operation such as publish or display a grid for alignment or layout of the sheet. The general classes of Active Objects are labels, quotes, tickers, time graphs, data set

i graphs, page segments and buttons. The tools may also be invoked by name rather than graphically by selecting an icon representing that tool using a pointing device. Further description of the various tools will be provided below.

The user programmable display area proper is shown at 14. At the top of the display area in this example, the user has created an instance of a label class Active Object with text entered by the user to define what the particular sheet being displayed is. This label Active Object was created by invoking the label tool represented by icon 15.

The program according to the teachings of the invention use objected oriented programming style. Although the preferred embodiment of a program according to the teachings of the invention has been written in C language for easier portability among machines using programming conventions to make the C language act like an object oriented programming language, it is easier to construct the program using object oriented programming languages such as C++.

The middle of the display area contains several instances of quote class Active Objects in different display formats generated by invoking the quote tool represented by icon 17. Each three, four, or five letter individual symbol in the box 18 and the number next to it represents an individual quote where the user specified the symbol based upon the stock issue he or she wanted to monitor.

The middle of the display also shows an instance of

- 11 -

a ticker class Active Object showing all trades in a specified set of issues that exceed a minimum volume set by the user. This particular criteria for display was programmed by the user using the ticker tool represented by icon 19.

At the bottom of the display is an instance 22 of a graph class of Active Object. The program allows graphing of trades, volumes, spreads, indexes or any other value. Two different types of graph tools are provided although only graph icon 21 is shown in Figure 1. In the preferred embodiment, another graph icon is present in area 13 representing a tool to create Active Object instances of data set graphs. These types of graphs are usually bar graphs where each bar represents some real time value about a different stock issue designated by the user although in different embodiments, each bar can represent the real time value of any variable in a complex system.

An instance of a user designated portion of a standard financial page such as Telerate, page 8 has been created at icon 24 using the page fragment tool represented by icon 26.

"Active document" as that term is used herein means a video displayed document of one or more "sheets" of the user's design which incorporates text, displays of real time data in user definable style, e.g., color, font, background, pen size etc. and format, e.g., quote, ticker, graph etc., alarms, and alarm scripts, i.e., user defined scripts of commands to be processed (much like a word processing or spreadsheet macro) when an alarm limit is exceeded. The program automatically accesses the network to which the host is connected through network interface software which establishes the proper subscriptions for the desired real time data with the appropriate information service and the server upon which this service process is running. Real time data is then passed to the program from whatever network communication process is being used and is immediately displayed in the format, style and location previously specified by the user. Although the invention

- 12 -

will hereafter be described in the preferred embodiment for use in a financial environment such as a trading floor of a broker such as Dean Witter etc., the invention is not limited to such applications. Any complex system which generates real time data which control operators must monitor are subject to being monitored and controlled using the teachings of the invention.

The attributes of the Active Objects which are used to compose an active document are as follows.

10       A label Active Object is just a fixed character string placed in a position on a sheet entered by the user. Labels are used to identify sheets, regions on sheets, and individual monitoring Active Objects as well as in script files to generate messages when an alarm event occurs or to generate customized help screen for a particular active document. A label does not change in real time. Its attributes are: String (field) which is the text string to be displayed in the label object; and, Alignment (radio button list) which is one of three formatting options - 20 left, center and right.

A quote object is a familiar market price quotation. Any number of quotes may be placed on a sheet. Attributes of the particular instance record for a quote Active Object determine not only what issue is quoted, but also what items 25 of information from the total comprehensive record for a particular issue are displayed. Different types of quotes may have different formats. The attributes of a quote are:

- **Market Type (list)**  
Defines the type of information being received from the source in this quote, such as equity, option, future or shredder output. This choice is used to determine the information source, the available styles and the internal record formats.
- **Display Style (list)**  
Defines what display format is to be used for this quote. Each market type has one or more styles

- 13 -

defined, which typically show different collections of fields from the received information. Most market types have a "brief" or "regular" style, which shows simply symbol and price information, and other styles may also be present which show more detail.

- **Symbol (field)**

Defines the specific symbol, i.e., stock, to be used for this quote instance. The symbol may have a suffix which indicates the exchange to use. These suffixes may be translated into a longer exchange name through translations set up by the System Administrator. Also note that for user convenience, character entered will automatically be put into upper case (if lower case letters are needed, such as for a class or warrant, the shift key can be used).

- **Alert Field (command button)**

Pops up a dialog box containing a list of numeric fields in records for this market type. Used to select which field to use in alert calculations. This dialog box also contains areas to enter four different scripts, which are executed under different conditions, as explained below.

- **Limits (2 buttons and fields)**

These are used to set above and below limits for an alert. Also, the above and below limits can be activated and deactivated by clicking on the check boxes. When the value of the real time data goes above or below the limits, the object is shown in reverse video.

The choices listed in the Market Type and Style lists are defined in configuration files. Other types and styles can be set up by the System Administrator.

Quote objects are traversable, which means that the user can transverse to the quote object and enter a new symbol simply by entering it from the keyboard.

- 14 -

Quotes have four different scripts which are run for different reasons with respect to alerts. Figure 2 illustrates states behind this concept. A quote is either in the normal state 26 or the alert state 28. When a real time data update comes into a normal state quote and does not trigger an alert, the "normal update" script is run. A script is a user defined string of commands that are executed in sequence. They can be commands that the program of the invention understands, commands to the operating system or other processes operating in the environment or commands to any other process running anywhere else on the network. The script language also has a glossary facility whereby the user can define new commands and add them to the script language. When an update comes in which triggers an alert, the "begin alert" script 30 is run. This script takes the quote object into the alert state 28 where the "alert update" script is run. Additional updates which are in the alert run will continue to run the "alert update" script. Then when an update comes in which is back in the normal range, the "end alert" script will be run, followed by the "normal update script. Thus, the four scripts provide a way of checking for changes in the state, or for staying in the same state.

The ticker tool is a continuously shifting display of trades in a specified list of issues. In addition to attributes, the dialog box for a ticker displays the current list of securities being tracked and some commands for manipulating the list. The dialog box is used to change or add to the securities on the list. The ticker attributes are:

- **Create (command button)**

Adds a security to the list. A mouse click on a Create Button, i.e., the icon 19 in Figure 1, will cause the Subscription Entry dialog box to appear. The subscription for the desired real time data is edited using the dialog box and is completed with

35



- 15 -

a mouse click on the OK button. Another subscription can be entered with another click on the Ticker icon 19 .

- **Delete (command button)**  
5 Deletes the currently selected item from the list.
- **Edit (command button)**  
To edit items which are already on the list, one pointing device click on the item selects it. Then a click on the Edit button, and the  
10 Subscription entry dialog box will appear.
- **Copy From (command button)**  
Replaces the current ticker list with a copy of the one from another ticker. A dialog box will pop up requesting the name of the source ticker.
- 15 • **Add From (command button)**  
Like Copy From but adds to the current ticker list instead of replacing it.
- **Name Sort (button)**  
Sorts the securities listed in the selection list  
20 by their symbol names.
- **Type Sort (button)**  
Sorts the securities listed in the selection list by their market type.

The subscription entry dialog for the ticker tool is as  
25 follows:

- **Market Type (list)**  
Used to select the Market Type for the security.
- **Ticker Style (list)**  
Used to select the display format for trades or  
30 updates to the ticker subscription instance. There are generally several styles, similar to those defined for the Quote object. The styles are generally different for the different market types.
- 35 • **Symbol (field)**  
Used to enter the security symbol. The same conventions are used as for entering the symbol

- 16. -

into the Quote dialog box.

- **Volume (field)**

5           the minimum number of shares in trades to be displayed. Note that this dialog box field will only be shown for Market Types which have an associated volume field. For instance, most equity fields have a volume but RTSS results do not.

Composite or "wild card" tickers can be created simply  
10 by entering for the symbol a period followed by the name of the exchange code. The ticker object will then show every update reported by the feed on that exchange.

The ticker object will show new data each time it receives an update from the data feed which includes either  
15 a new volume value, or a new value for one of the fields in the specified style. Thus, if the display style shows the symbol, the last price, and the trade volume, updates will appear when there is a change of the last price or the trade volume. If the display style shows the symbol and the bid  
20 and ask price, updates will appear when there is a change of the bid price, ask price, or the volume field. In this way, the ticker can handle information from sources which do not have the standard field, such as output from the Shredder, an application that elementizes data which is provided by a  
25 service only in standard "packages" such as Telerate pages etc.

The Page Fragment tool, represented in Figure 1 by the icon 26, is a user selectable rectangular section of a page from a page-based market data service. The user constructs  
30 a page fragment interactively, using a pointing device to designate the fragment to be extracted and its destination on the sheet being composed. This can be done starting from a page-based service application, or entirely within the program of the invention. The user can then modify these  
35 and other attributes using the page fragment dialog box and the menu commands.

- 17 -

To make a page fragment Active Object to be displayed, the following process is performed:

1. Move the cursor to a window, i.e., a page-based display, containing the source page.
- 5     2. Position the cursor at the upper left corner of the region the user wish to select.
3. Click and hold down the left mouse button and move the cursor to the right and down to the lower right corner of the region the user wishes to select. The fragment can be as small as two characters or as large as the entire page.
- 10    4. Release the mouse button. The fragment will appear in reverse video.
5. Select the Page Fragment tool from the toolbox by clicking on icon 26 or typing the name of the tool or using any other input device to select the tool.
- 15    6. Click the mouse button where the page fragment is to be located on the sheet being composed and drag to the lower right.
- 20

To make a page fragment with no source page:

1. Select the Page Fragment tool from the toolbox.
2. Drag the region where the page fragment is to be positioned on the sheet being composed.
- 25    3. The page fragment will be blank except for the title. Use the dialog box to define the service, page and region to be displayed in the page fragment. The default region will be the full page.

30     The dialog box for a page fragment will display the full source page, and draw the region being extracted for the fragment. The program according to the teachings of the invention draws display objects by invoking the functions of the commercially available XWINDOWS system in 5 the preferred embodiment although other embodiments can be  
35     ported to any other commercially available or research

- 18 -

screen rendering library, program or toolbox. The XWINDOWS system is available from a consortium of universities and commercial vendors known to those skilled in the art. To change the region in which the page fragment is displayed, the mouse or pointing device can be used to designate the upper left corner of the new region, and then drag to the lower right corner. The dialog box can also be used to set the following attributes:

- **Service (field)**  
10 The name of the page-based source, such as Telerate. Click on this field to bring up a dialog box to change the service. If another page fragment is also displaying that service and page, it will be immediately shown in full source page display. If not,  
15 the user must confirm the dialog box to request the page from the service.
- **Page (field)**  
Page Number (or name), such as 7677. Click on this field to bring up a dialog box to change the page. If  
20 another page fragment is also displaying that service and page, it will be immediately shown in full source page display. If not, the user must confirm the dialog box to request the page from the service.
- **Title Position (radio button list)**  
25 Used to specify that there should be no title, or that there should be a title indicating the service and page number, and the highlighting mode. The default is that there is no title, in order to pack page fragments closely together.
- 30 Page Fragments are traversable. The user can traverse to a page fragment object, and enter a new page name or number simply by typing it from the keyboard.

The Graph tool allows creation of graph Active Objects with associated display objects which track the real time  
35 variations in a variable in a complex system such as market value. The graph object is a composite of several simpler objects: a graph/time axis, a price axis, and a graph area.

- 19 -

Each of these areas can be independently selected for changes to colors, fonts, and borders. The outermost portion is the frame for all of the graph sub-objects, and must be selected in order to perform edit operations such as 5 move, resize, or front/back restacking operations.

Each axis has its own attributes controlled by a separate dialog box. Because the graph object is a composite object, it may not be duplicated with the Duplicate tool.

When the user clicks and holds a mouse button anywhere 10 inside the graph area, the value for the Price Axis is shown in the upper left hand corner of the Graph Display Area and the value for the Time Axis is shown in the upper right hand corner. This feature is useful in identifying numeric values for any point on the Graph Lines.

15 The Graph/Time axis dialog box allows the user to edit attributes of the overall graph as well as the attributes of the times axis. The attributes this dialog box currently supports are divided into the following categories.

#### **SCALING ATTRIBUTES**

20 • **Type (radio button list)**

Selects the type of time axis. The time axis currently support two types of scales: time and ticks. The default is time.

• **Precision (radio button list)**

25 When the scale type is set to time, the precision of the scale mark labels can be set to minutes or seconds. When the scale is set to ticks, precision is not adjustable.

• **Lock Scale (option button)**

30 freezes the time axis minimum and maximum values to their current settings, disabling automatic rescaling. When this option has been selected, a lock icon will be shown on the axis.

#### **TIME AXIS LIMIT ATTRIBUTES**

35 • **Maximum (field)**

- 20 -

- The upper limit of the time scale.
- **Minimum (field)**  
The lower limit of the time scale.
- **Step Value (field)**  
5 The increment to show when drawing hash marks and numbers on the time axis. Axis limits can be changed manually in the dialog box or can be changed using scaling commands
- **Draw Grid Lines (option button)**  
10 If selected, the tick marks on the times axis are drawn across the graph area.

#### CHARACTERISTICS

- Axis Position (radio button list): selects position of time axis as either above or below the graph area.
- 15 • **New Price Axis (command button)**  
creates a new additional price axis and positions it to the right of the existing axis.

#### GRAPH PRICE AXIS

- Each graph can one or more price axis, each  
20 corresponding to a different instrument being graphed.

#### NAME

- **Symbol Name (field)**  
Defines the specific symbol to be used for this price axis, such as IBM or GOLD. This may be followed by an  
25 exchange code. The same conventions used in the Quote symbol name apply here.

#### PRICE AXIS LIMITS

- **Max Value (field)**  
Upper limit of price scale.
- 30 • **Min Value (field)**  
Low limit of price scale.
- **Step Value (field)**  
The increment to show when drawing hash marks and

- 21 -

numbers on the price axis. Axis limits can be changed manually in the dialog box or can be changed using scaling commands.

- **Draw Grid Lines (option button)**
- 5 If selected, the tick marks on the time axis are drawn across the graph area.

#### **PRICE AXIS COMMANDS**

- **Delete Axis (command button)**  
Deletes the selected price axis unless it is the only price axis.
- 10 • **Copy Axis (command button)**  
copies the selected price axis and places it on the screen to the right of the existing axes
- **Clear Trends (command button)**
- 15 removes all trend lines related to the current price axis

#### **Price Axis Attributes**

- **Line Width (field)**  
determines the width of the Graph Plot Line. Width of 0 is recommended for optimum repainting time .
- 20 • **Retain Count (field)**  
the number of data points that are kept for the given axis. The default is 150, this means that the system keeps track of the last 150 "ticks". Once 151 is plotted, the first tick is dropped off.
- 25 • **Point Size (field)**  
the size of the Point drawn when a new data value is plotted.
- **Precision (field)**
- 30 the number of decimal places to show on the Price Axis (Y-axis) has marks.
- **Connect Points (radio button list)**  
If "Yes", will draw lines between data points; If "No", will only plot data points (be sure to set Point Size to greater than 1 if lines aren't drawn) .
- 35 • **Direction (radio button list)**

- 22 -

places the current Price Axis on the Left or Right hand side (default is left).

- **Lock Scale (option button)**  
freezes the price axis minimum and maximum values to their current settings, disabling automatic rescaling. When this option has been selected, a lock icon will be shown on the axis.

#### **GRAPH TREND LINES**

Trend Lines may be drawn on a graph to indicate support and resistance levels.

#### **CREATING TREND LINES**

1. Move the cursor to the point on the Graph where the Trend Line should begin.
2. Hold the Shift Key down and click the left mouse button. This creates one trend point. The user can move the trend point before releasing the button by moving the mouse to the desired location.
3. Click the left mouse button a second time on the first trend point, hold it down, and drag the cursor to a second trend point. The user will see a temporary trend line "following" the mouse cursor as the user drags it into position.
4. Release the Mouse Button to complete the line. It will appear as a dotted line.

#### **EDITING TREND LINES**

- **Insert Trend Points/Segments**  
Press and hold the left mouse button on one of the trend points. A new trend point will be created that can be dragged to the desired location.
- **Move Trend Point**  
Press and hold the right mouse button on the trend point to be moved and drag it to the desired location.
- **Delete Trend Point**  
Hold the shift key down, and click on the trend point with the right mouse button.



- 23 -

- **Move Trend Line**

Hold down the Shift and Control keys together. Press and hold the right mouse button on any trend point on the line. Then drag the trend line to the desired location.

**DELETING TREND LINES**

Hold down the Shift and Control keys together and click the left mouse button on and trend point on the line. All trend lines can be deleted by using the clear trends command in the Price Axis Dialog Box.

**GRAPH SCALING**

The dynamic graph application provides very flexible scaling functions as follows:

- **Settable min, max, and step values**

The minimum, maximum, and step values of the axes can be set via the dialog boxes discussed in the Price Axis sections. They can also be changed via the rescaling methods described below (zooming, and moving trend lines).

- **Automatic Rescaling**

Axes can be rescaled automatically when new data points arrive. Unless scale is fixed, the graph adjusts periodically to accommodate the current shape of the line.

- **Fixed Scaling**

The automatic rescaling feature can be turned off by setting the lock scale attribute to true. This attribute can be set via the dialog box.

- **Zoom Function**

The graph object provides the capability to zoom in on a rectangular section of the graph. The same action can be accomplished by setting the axis limits manually, but it is much easier to perform such an operation using the mouse. To initiate a zoom, hold down the CONTROL key, click the LEFT mouse button at one corner of the desired zooming rectangle, drag the

- 24 -

5 cursor to the opposite corner of the rectangle, and then release the button. The axes will rescale such that the specified rectangular viewing region takes up the whole graph viewing region. To zoom back such that all points are visible, hold down the CONTROL key and click the RIGHT mouse button anywhere on the graph view area.

- **Rescaling via trend line drawing**

10 There is a potentially useful side effect of drawing trend lines that can be utilized as a quick method for rescaling the axes to the desired values. When dragging a trend line point, the user may move the point outside of the visible graph window. If the user releases the button while the point is outside, and the  
15 appropriate axes are not "fixed," then the axes will rescale automatically to accommodate the new point within the graph area. Thus to extend the time axis further to the right, instead of changing the max value via the dialog box, the user may move a trend line  
20 point the desired distance beyond the right boundary of the graph view. When the user releases the mouse button, the time axis will extend its scaling to include the new trend line point.

#### **GRAPH ALERTS**

25 The graph alert facility provides a notification if a data point moves outside a region defined by two active trend lines. At any given time, at most two trend lines can be active. If two trends are active, then activating a new trend line will deactivate the last recently activated trend  
30 line of the two. Activation and deactivation of trend lines is done via an abbreviated "move" trend point operation, i.e. clicking on any point of the trend line with the right mouse button without any mouse movement between press and release. Clicking on an inactive trend line activates it  
35 while clicking on an active trend line will deactivate it. An active trend line is drawn in the color of the graph plot

- 25 -

with which it is associated. Inactive trend lines appear in a lighter color.

If two trend lines are active, any new point that comes in through the subscription will be checked to see if it falls between the values specified within the time range of the two trend lines. An alert is triggered when a graphed value goes outside the bounds of the active trend lines.

To use the alert facility:

1. Make sure the correct price axis is highlighted for the security for which the user wishes to set an alert.
2. Create two trend lines that define upper and lower limit ranges (a "channel") for the security.
3. Activate the trend lines by clicking on any trend point with the right mouse button. An active trend line is shown as a dotted line the color of the graph plot line. An inactive trend line is shown as a lightly colored line. The right mouse button is a toggle between active and inactive.
4. If the security value goes above or below the channel formed by the two trend lines, the graph will alert. The alert is shown by the new trend point flashing as a large square.
5. The alert will continue to flash until a new trend point alerts, or if the user clicks on the alert with the left mouse button.

There are two scripts associated with graph objects, one for normal updates and one for alert updates.

#### **BUTTON**

A button is an object which the user can interact with, and will cause a script to be carried out when clicked on. For instance, the button might perform the equivalent of a Sheet Next command, or transfer the user to a specific sheet. Buttons allow the user to determine the dynamic

- 26 -

action of the sheets, as well as their appearance. These scripts are expressed in the 5 MarketScript® language.

Each button is created using the button tool, and the user can use the font fill, and pen menus to specify the appearance of the button (the border menu does not operate on buttons). However, since a button carries out its operation when clicked on with the left mouse button, the user must use the middle or right mouse buttons to bring up the dialog box to set the button attributes. The "Script" entry of the button dialog box allows the user to enter the script to be carried out.

#### BUTTON STYLES

Style include rectangular, rounded, shadow, check box, and radio button. In most cases, the check box and radio button styles are used when there are groups of buttons. Note that for check box and radio button styles, the alignment choice does not apply.

#### SCRIPTING LANGUAGE

Scripts are composed of one or more actions, which are entered as a sequence. The actions can perform operations such as selection of objects, editing of selected objects, navigation to sheets, and editing of the current sheet. Note that these commands must be entered exactly as shown with respect to case. The currently supported scripting commands are:

##### Object Selection Commands:

	all select	selects all objects on current sheet
	none select	selects no objects on current sheet
30	self select	selects the object which is running the script
	<item_name> find select	selects all items of given name
35	<class_name> class select	selects all times of given class
	<region_bounds> region select	selects all items in region

- 27 -

## Editing of Selected Items:

	front	restacks selected items to front
	back	restacks selected items to back
5	hide	hides selected items
	show	shows selected items
	clear highlight	clear highlight of selected page fragment objects
10	<color_name> pen	sets pen color of selected items
	<color_name> border Color	sets border color of selected items
	<integer> borderWidth	sets border width of selected items
15	<color_name> fill	sets fill color of selected items
	toggle visibility	inverts hide/show of selected items

## 20 Move and Resize of Selected Items:

	<xY> move	move of selected items to x, y
	<dxXdy> move	relative move of selected items
25	<widthXheight> size	resizes selected items to width, height
	<dwidthXdheight> rsize	relative resize of selected items
	<x> alignLeft	aligns left edge of selected items to x
30	<y> alignRight	aligns right edge of selected items to x
	<y> alignTop	aligns top edge of selected items to y
35	<y> alignBottom	aligns bottom edge of selected items to y

## Sheet Navigation Commands:

	first sheet	transfer to first sheet
	prev sheet	transfer to previous sheet
40	next sheet	transfer to next sheet
	last sheet	transfer to last sheet
	index sheet	active index dialog
	<sheet_name> find sheet	transfers to next sheet with given name. wraps

- 28 -

	push sheet	back to front if needed. beeps if name not found. pushes current sheet onto stack
5	pop sheet	pops entry off of sheet stack and goes to that sheet
 Editing of Current Sheet:		
	<color_name> background	sets sheet color background to named color
10	on gridLines	turns on drawing of gridlines
	off gridLines	turns off drawing of gridlines
15	toggle gridLines	toggles the drawing of gridlines
 Execution Control		
	beep	beeps the workstation
	<UNIX command> execute	runs UNIX command in background
20	<milliseconds> pause	pause for specified amount of time
 Environment Control:		
	on toolBox	shows the toolbox
25	off toolBox	hides the toolbox
	toggle toolBox	inverts hide/show sense of toolbox
	on gridTool	turns on grid tool
	off gridTool	turns off grid tool
30	toggle gridTool	toggles activation of grid tool

**SCRIPTING EXAMPLES**

You will find the sheet operations most useful when creating new traversal options on a set of sheets. The  
 35 "find sheet" command is most useful to transfer to a specific sheet, and will search from the first sheet to the last to find one which matches the give name.

The object selection commands provide ways to select all objects, no objects, all of a given class, all of a  
 40 given name, or all in a region. For instance, if the user created a set of Quote objects, and had some display elements of the Dow-Jones Industrials and the others display

- 29 -

the Dow-Jones Utilities, assign the name "INDU" to all of one group, and assign the name "UTIL" to the other group.

Then use a button script of "INDU find select red fill" to turn all of the Industrials red, and "UTIL find select 5 blue fill" to turn all of the Utilities blue under button control. Another useful example would be to use "Quote class select green fill" to change all quotes to have a green fill color.

A common use of scripts and pop-up objects is to 10 annotate a sheet with a number of extra label objects which describe what is being monitored or graphed. Make sure that the label objects are in front of the other objects, and then give them all the name "DETAIL". Now create a button with the script "DETAIL find select show", and another with 15 the script "DETAIL find select hide". Executing these two buttons will turn the detail on and off on your sheet.

#### SCRIPTING NOTES

If an item name, sheet name, command, or color name contains a space, it must be contained within quotation 20 marks. Either single or double quotes may be used. Names containing single quotes may be contained in double quotes, and vice versa.

When setting up alert scripts for a group of Quotes, set up the scripts on the first one, and then use the 25 Duplicate tool to make the others. The scripts will be copied into each one made.

There are roughly 80 pixels per inch on a workstation screen. All of the x, y, width, and height measures in the scripting language are in pixels.

30 It is common for scripts to select objects to operate on, but if scripts are run at unexpected times (such as due to Quote updates), they could change the selection status of the objects the user is trying to edit. For this reason, the selection status of the objects is saved at the start of 35 the script execution, and is restored afterwards. Thus if a script begins with an "all select" phrase, it is not

- 30 -

necessary to end with a "none select phrase".

Referring to Figure 3, there is shown a typical network environment in which the preferred embodiment of the invention would be employed. Elementized data feeds are received via a LAN/WAN (Local Area Network/Wide Area Network) 32 and handled by a feed handler 34. Paged market feeds are received via the network 32 and handled by a feed handler 36. A workstation 38 running a shredder process breaks up the pages of the paged feeds into their constituent elements. A workstation 40 running a program according to the teachings of the invention is used by a user to compose an Active Document to manage financial information in real time. As the various quotes, tickers, graphs etc. are created, subscription requests for the required data are passed to the network communication software running on the workstation 40. If the TIB® network communication software is being used, the subscription requests are filled using subject based addressing. The subscription requests result in properly formatted service request transmissions on the network 32 to the proper service and server in the proper protocol to request the desired data. The server or servers then transmit the data back to the workstation 40 where it is displayed in the Active Document defined by the user.

Referring to Figure 4, there is shown a typical example of an "object" as that term is used in the object oriented programming world. An object is an entity which has both properties and associated operations which can be invoked by a user to change the values of various properties of the object. In the simple example chosen, the object is an entity representing a class of objects in the form of bank accounts. The particular instance in this class is a bank account for Jessica Doe. The properties of each object in this class are the account number, the balance and the owner name. The operations that can be invoked are withdrawal, deposit and query. Each operation can be invoked by entering a command to identify the object and start the



- 31 -

operation (typically the name of the operation) followed by the argument. The operation then uses the argument to change the properties of the object. For example, the command Doe.Deposit(100) would invoke the deposit operation 5 and add 100 units to the balance property of the Doe bank account.

Typically, an object is represented in memory by a record containing the values for each of the properties and a link or pointer to a library of programs which carry out 10 the operations associated with that object. This structure is shown in Figure 5. In Figure 5, two data records 42 and 44 represent the values of the properties for two instance of bank account class objects. Each record has a link or pointer, 46 and 48, respectively, to a library 50 of 15 programs which carry out the operations associated with this class of objects.

Referring to Figure 6, there is shown a diagram of the hierarchy of the parts of a typical program according to the teachings of the invention. A Frame Object 52 "contains" 20 all the other Active Objects that will be manipulated by the system. The Frame Object contains menus, dialog boxes and exactly one Active Document. The menus are comprised of a menu system 54 and a plurality of menu objects shown generally at 56. The menu system 54 handles user inputs 25 from whatever input device is being used for selection of commands, i.e., menu options. The dialog boxes are implemented by a dialog manager 58 and a series of dialog objects shown generally at 60. The dialog manager 58 handles user input associated with user selection of desired 30 values for editing properties of Active Object instances which have been created on a Sheet using a tool. It is also used to specify commands which are specific to certain types of Active Objects. Generally, the menu options shown at 12 in Figure 1 are generic operations which can be applied to 35 any Active Object. However, certain Active Objects have associated properties and operations which are specific to that class of Active Object alone. These specific

- 32 -

properties and operations are embodied in the instances of dialog objects shown at 60 in Figure 6. For each class of Active Object which has specific properties and operations, there is one or more dialog object(s) which can be used in  
5 a user dialog to set specific properties of an instance of that class of Active Object and to invoke operations which are specific to that class of Active Object. A graph class of Active Object is an example of a type of Active Object which has more than one dialog box. There is one dialog box  
10 to set the properties of the x-axis and one dialog box to set the properties of the Y-axis. Some of these dialog boxes "pop-up" supplementary dialog boxes.

An Active Document object 62 "contains" a plurality of other objects. They are: a plurality of Sheet Objects which  
15 have been composed by the user using the tools described above and shown generally at 64; a plurality of Active Objects the instances of which are created using the tools described above and shown generally at 66; and group objects shown generally at 68. Group objects are essentially  
20 "containers" for other objects and can contain a plurality of Active Objects such as are shown generally at 70 instances of which are created using the tools, and one or more other group objects shown generally at 72 each of which may contain a plurality of Active Objects such as shown at  
25 74 or other group objects such as are shown at 76.

Some of the properties of each Sheet Object are: whether a grid used for alignment of various Active Objects thereon is or is not displayed; grid size; background fill pattern, e.g., solid blue, dollar signs, etc.; a list of  
30 contained objects; scripts associated with the Sheet that handle alerts for any objects on the Sheet; whether the grid is being used; edit mode status, i.e., whether the user will be allowed to edit (unlocked) or not be allowed to edit (locked) the Active Document; etc. Some of the  
35 operations/commands associated with each Sheet Object are: toggle grid setting; change grid size; change background fill pattern; align an Active Object to the grid; change the

- 33 -

edit mode status; etc.

The Active Document object 62 has various properties including: the current Sheet to display; the list of Sheets in the Active Document; the file name under which the Active Document is stored; the number of Sheets; etc. Some of the associated operations include: open an Active Document file; add a Sheet or Sheets from another Active Document to this Active Document; close an Active Document file with save or do not save options; go to next Sheet; go to previous Sheet; etc.

The properties and operations associated with each Active Object on a Sheet are as defined above in the descriptions of Active Objects created with the tools.

The pop-up or pull-down menus under the main menu options of the menu bar are menu objects as are the main menu options of menu bar 12 in Figure 1.

The dialog manager 58 insures that the proper dialog object is selected to make the proper dialog box appear when a tool is invoked which requires a dialog to set the attributes of an instance of an Active Object being placed on a Sheet by a user. The dialog objects are the objects which actually embody the dialog boxes. The dialog objects contain text attributes the define the user's options and associated operations or commands which the user can invoke, e.g., graphically by clicking on an icon representing that operation, to change the properties or attributes of the Active Object to which the dialog box is associated.

The Active Objects such as those shown at 66 consist of labels, quotes, tickers, buttons, graphs etc. label objects are "active" even though they are text because they can, for example, be changed as part of a script of commands which are executed as a result of an alarm condition.

All the objects on the right side of the line 78 are stored in memory using the same structure as any Active Object with properties and associated operations. Some properties or operations may be nullities however so that the same structure can be used for all types of objects to

- 34 -

the right of line 80.

A tools manager 78 manages a plurality of tools objects shown generally at 80. The tools manager 78 handles the user input events which are associated with selection of the class of the next type of Active Object to be created. It is also used to edit the existing Active Objects. The tools objects have associated Display Objects which invoke the display/windows system through the applications interface 82 to render the icons which represent each tool. In the preferred embodiment, the XWindow system is used to render the icons which represent each tool. Each tool object has operations associated therewith which implement the operations defined above in the descriptions of the tools. The XWindow system is available commercially from the XWindow consortium of universities headed by MIT. It is also available through most workstation vendors such as Sun Microsystems. In other embodiments other window systems such as Microsoft Windows 3.0 or the Macintosh toolbox in the Macintosh operating system may also be used, or a screen rendering system can be programmed in from scratch.

A global event dispatcher 84 receives event information such as real time data updates from subscriptions and user input events and dispatches the data to the appropriate object or manager or other portion of the system to cause appropriate processing.

A script processor 86 interprets the commands of scripts entered by a user defining the desired processing to be performed in the case where an Event Triggering that script has occurred. Such an event could be the activation of a button or a real time data update which exceeds an alarm limit programmed by the user. Basically, the script processor handles requests to process scripts generated by the instances of the Active Objects programmed onto the various Sheets by the user.

Program systems according to the teachings of the invention operate in an "environment" comprised of various other support processes (programs in execution on the host

- 35 -

computer). The environment for a system according to the teachings of the invention is shown below the line 88. A high end network interface 90 receives requests for data from the system and communicates on the network (not shown) 5 to obtain the requested data. In the preferred embodiment, the network interface is the TIB<sup>®</sup> network communication software commercially available from Teknekron Software Systems in Palo Alto, California, but other network communication software may also be used. For example, the 10 IBM Datatrade<sup>®</sup> system, DEC's DECTrade<sup>®</sup> system or other high level network interfaces developed in the future. The TIB software automatically maps subscription requests on particular subjects, e.g., IBM equity prices, into the appropriate network addresses for the appropriate service 15 and server that provides this information and selects and carries out the appropriate service discipline and network protocols to communicate with the selected server over the particular network LAN/WAN system to which the host running the system according to the invention is connected.

20 An operating system 92 is also part of the environment and carries out host traffic direction and computer resource management duties necessary to support the system according to the teachings of the invention.

A file application programmers interface and file 25 management process 94 is also part of the environment and typically runs on a file server, although in some embodiments, it may be part of the operating system.

Finally, the environment includes the display/window systems application programmer's interface for screen 30 rendering. The preferred embodiment of a system according to the teachings of the invention utilizes the commercially available XWindow screen rendering system. This system can receive commands to draw various objects such as line, point, box or other primitives and text, font and color 35 codes etc. from an application program and then render the screen accordingly. Any known screen rendering system can be used in other embodiments, or the screen rendering function

- 36 -

can be programmed as part of the program system according to the teachings of the invention.

Referring to Figure 7, there is shown a diagram of the parts of an Active Object. An instance of an Active Object 5 100 contains a data object 102, a Style Map 104, a Display Object 106, one or more Event Triggers shown generally at 108 and one or more Event Scripts of commands to be performed sequentially if an Event Trigger condition occurs. These Event Scripts are shown generally at 110. The Active 10 Object 100 includes a local event dispatcher which dispatches events, e.g., user commands, to the various Active Object components for processing. For example, data updates from a network or database are dispatched to the Data Object 102, and "display frame" or display related 15 events are dispatched to the Display Object 106. Events handled by the local dispatcher can be internally generated, such as a "change notification" event from the Data Object, which is dispatched to the Display Object so that the displayed representation can be updated.

20 The Data Object contains the internal representation of the data being manipulated by the Active Object, i.e., a full listing of all the data which is available about a particular subject, only some of which does the user desire to view. For example, typically a quote object will have a 25 Data Object with a large number of fields of data about a particular company and the trading activity in that company's stock.

The Style Map 104 contains a human readable specification defined by the user via selections from a 30 dialog box which popped up when the instance of the quote Active Object was created which specify which of the data fields in the Data Object to display in a display frame or box shown on the display to represent the Active Object and where to display the selected fields in the display frame. 35 The Style Map also contains data which specifies certain display characteristics such as what color to display the object when the price is going up and what color when the

- 37 -

price is going down. The Style Map is simply a specification. It does not do any of the actual work of rendering the Active Object instance or screening and selecting data from the Data Object to be displayed.

5 The Active Object 100 tells the Display Object 106 what Style Map to use. Then a style processor (not shown) in 25 the Display Object does the work of extracting the proper data from the Data Object and sending the proper commands to the screen rendering system to display the desired data in 10 the user selected format and style. The Display Object 106 encapsulates the data and operations required to display the data object in the "style" requested by the user. That is, the Display Object contains the internal representation of the actual data to be displayed for this Active Object in 15 the location on the current Sheet specified by the user and a style processor. This internal representation is sent to the screen rendering system by the style processor to actually draw the display seen by the user. The style processor is actually implemented in a library and the 20 Display Object 106 contains a pointer to this library and receives a pointer to the style map 104 such that the Data Object can be processed by the style processor library programs in 5 accordance with the style map.

The Event Trigger is a specification of conditions 25 under which the user wishes to do extra processing on the Active Object. For example, the user can set alarm limits such as a certain price or trading volume for a particular quote Active Object, and when a real time data update indicate that the limit has been exceeded, an alarm 30 condition exits to transfer the Active Object from the normal update state to the alert state. The Event Script of commands to execute upon occurrence of the specified alarm condition is specified in the Event Script specifications shown generally at 110. The things that can be scripted to 35 happen upon occurrence of an alarm condition are limited only by the imagination of the user. Minimally, the script may specify an audible beep and/or a change in color of an

- 38 -

Active Object. More exotic scripts may issue commands on the network to start another process running to dial a beeper, issue a sell order, issue a buy order, etc. Other scripts may publish some or all the data on one or more  
5 sheets of an active document on the network, etc. The commands in the scripting language generally include all the commands understood by the script processor as well as commands defined by the user and can, in some embodiments, include commands to the operating system, the high level  
10 network interface or other processes running on the network. Generally the commands understood by the script processor will include the name of the object, the desired operation and an argument, i.e., what value to set etc.

The Active Object 100 also has associated operations  
15 such as "move yourself", "change the data using the real time network data update or a database update"; etc. Most operations are done by the local dispatcher. For example, data update events cause the local dispatcher to send the data to the Data Object 102. Programs according to the  
20 teachings of the invention are written for "event driven" processing style. An "event" as that term is used here can mean a user caused event such as an input from the keyboard, pointing device or other input device or it could be generated by the program itself, the network interface, or  
25 any other process in the environment or elsewhere on the network. The event dispatchers in the system match each event with some operation to call. The Active Objects have their own event dispatchers and there is a global event dispatcher which deals with input events from the network,  
30 the input devices, the operating system or a database which is used to store pricing information so that when a subscription is entered, the user may get the most recent data immediately and does not have to wait for the next update.

35 Referring to Figure 8, there is shown a flow chart of the processing performed by the style processor for each Active Object upon the occurrence of a data update event.



- 39 -

A data update event, represented at 112 causes the style processor in the Display Object 106 in step 114 to extract the values from the user specified fields from Data Object 102. The extracted values are then stored in step 116 in;  
5 the Display Object in accordance with the specifications set by the user in the Style Map 104. The value of the data fields are then compared to any event trigger specifications set by the user and stored in the Event Trigger specification 108. Step 118 represents the process of  
10 making this comparison and computing any special effects specified in the Style Map such as color changes, blinking etc. Step 118 may result in an update to the Display Object 106 if any special effects are triggered. Step 120 represents the process of notifying the parent Active Object  
15 to send the appropriate commands to the screen rendering system to render the Display Object 106 in the preferred embodiment. In other embodiments, the style processor itself can send the commands to the screen rendering system.

Referring to Figure 9, there is shown a diagram of the  
20 event architecture of a program according to the teachings of the invention. At the left are shown symbols for some of the input event generators. User events can be generated using a voice processor 124, a keyboard 126, a mouse 128, or a touchscreen 130 or any other user manipulated device.  
25 Other input events can originate from the network 32, a file/database 132 or from the operating system 134. The file/database stores real time data as it comes in from the network so that users requesting a price etc. can immediately have access to the latest information and need  
30 not wait for the next update of the requested information. An event originating from the network or the database is usually the data returned after a request generated by the creation of an Active Object such as a quote, ticker, graph etc.

35 A global dispatcher 136 receives the event and determines which object in the system it pertains to. The dispatcher then makes a call to the appropriate operation of

- 40 -

the appropriate object to process the event as symbolized by the lines of event flow 138.

After the environment is initialized, the global dispatcher 136 has control of the system. It then waits for  
5 an event and processes each event appropriately. For example, a mouse event in the form of a click on a frame object icon such as a window meaning move this window to the top of the stack will be dispatched to the frame object 52 by calling the appropriate operation to move this window to  
10 the top of the stack. A mouse click on a menu bar option as shown at 12 in Figure 1 would be dispatched to the menu manager 54 as a call to the operation to display the appropriate pop-up menu for the suboptions of the selected menu option.

15 Typically, the global dispatcher 136 will dispatch an event to the dialogue manager 58 when the user takes an input action to create an Active Object which requires a dialogue box. A typical sequence would be for the user to click the mouse on a tool to create an instance of an  
20 object. This event would be dispatched to the tools manager 78 which would select the appropriate one of the tool objects 80. The user would then move the mouse to where the instance of this class of object is to be displayed. This event would be dispatched to the appropriate tool object  
25 which would then create an instance of that type of Active Object at the desired location. A display object representing that object is created and commands are sent to the rendering system to draw the display object. The user then clicks the mouse on this displayed Active Object and  
30 the appropriate one of the dialogue boxes 60 pops up. The user may then select the various options in the dialogue box to invoke operations to set up the properties of the Active Object represented by one of the Active Object boxes at 66. After the user is done specifying the Active Object desired,  
35 the dialogue box performs an update operation when the save button is "pushed" to set the properties of the Active Object as specified by the user.

- 41 -

A network event is dispatched by the global dispatcher to a network data dispatcher 150. This dispatcher examines the type of data it is and sends it to the appropriate Active Object. The network data dispatcher can "fan" the data out to more than one Active Object if more than one Active Object exists with an interest in this type of data. A file/database data dispatcher 152 does the same thing as the network data dispatcher except it does it for data events originating from the file/database 132 after read I/O operation.

Some events can originate within and are locally dispatched from the Active Objects themselves or from the menu objects. For example, the user may click on a font change option from one of the menu objects 56, and this event will be dispatched as an update to the style map of the currently selected Active Object. Likewise, a data update to an Active Object may cause the value of some variable to exceed the alarm limits for that variable set by the user. In such a case, the local dispatcher of the Active Object making the transition into the alert state will invoke a script processor 154 and send the user specified script for the appropriate alarm event to the script processor. The script processor then processes the script to carry out the commands specified in the script in the order specified in the script. If one of the commands in the script is to change a color or a font, the script processor will call the style map of the Active Object specified in the script (it may be different than the Active Object which triggered the script processing) and update the style map of that Active Object. If the script calls for publishing some data, the style processor calls the high level network interface 90, invokes a publish function and sends the appropriate data to be published on the network. Likewise, if the script calls for sending data to a database 156, this also can be done. The script may also call for invoking some function of the operating system through an operating system call 158, and can invoke other applications

- 42 -

160 running in the same environment. Further, the script processor may also cause; the other application to perform some function and may even cause the other application to access the network through the network interface 162 of the  
5 other application to either send or receive some data specified by the script.

Of course the Active Objects all may dispatch rendering commands to the display rendering software interface to render their respective display objects.

10 The script processor 154 may also be called by the menu objects 56 or the dialogue boxes 60. This allows the; user to directly cause scripts previously defined by that user or some other user to be run by selection of a menu or dialogue box option to that effect and designating the desired script  
15 to run.

The objects of the Active Document are arranged in a hierarchy, and escalation of events within the hierarchy is allowed. For example, the line 164 represents the fact that the frame object manages the menu manager, and the line 166  
20 represents the fact that the menu manager manages the menu objects. An example of an escalation of an event would be where the user selects the close option to close the active document. This event would first be escalated from the appropriate menu object to the menu manager 54 to make the  
25 menu disappear and then would be escalated to the frame object 52 because the menu manager does not have an operation associated with it to close an active document. Only the frame object has that capability.

Similar hierarchy exists for the frame object, dialogue  
30 manager and dialogue boxes. A similar hierarchy exists within the Active Objects themselves although this is not shown. The Active Objects are also managed by sheet objects which are not shown in Figure 9 but which can be seen in Figure 6. The sheet objects are managed by the frame  
35 object. An example of escalation of an event in this hierarchy is when the user invokes a "delete yourself" command to delete an Active Object. The will cause an event

- 43 -

escalation from the selected Active Object to the corresponding sheet object on which the Active Object appeared to cause the sheet object to delete the Active Object from its list of current Active Objects.

5 Simultaneous event processing is allowed. For example, if the user is dragging an Active Object to another location with the mouse, a series of mouse move events are occurring which are being dispatched to the frame object 52 which contains the code for processing mouse moves. These events  
10 result in event dispatches to the selected Active Object to tell it to move itself. Simultaneously, a network update event can come in to cause a value in the Active Object to be changed and this may cause an alarm and the associated script processing.

15 In alternative embodiments, the tools may be represented as menu selections or simply be invoked by typing or speaking their names rather than through graphic selection of icons with a mouse, touchscreen, trackball or through keyboard navigation with the arrow or tab keys. In  
20 some embodiments, a map-like coordinate system could be used, and the user could select tools, menu options or Active Objects by speaking their coordinates. Their must be one input device however, although it may be connected to the host running the program according to the teachings of  
25 the invention through the network or through another application with an interprocess communication mechanism such as shared memory, Unix pipes, etc.

In some embodiments, the program itself may specify the placement of Active Objects and automatically fit them  
30 together after the user specifies what he or she wants in the active document.

Button objects provide great flexibility. A button object can be scripted to pop up on a user written help screen, make another Active Object appear or disappear, or  
35 do any other commands or sequence of commands within the scripting language.

The menu bar 12 in Figure 1 includes a hide/show toggle

- 44 -

option wherein Active Objects can be hidden, i.e., not displayed, although an internal representation of the display object is maintained so that at any time, the object may be displayed once again. This is especially helpful for user defined help screens which the user does not want visible at all times.

#### **MarketSheet™ APPLICATION**

MarketSheet™ is a Teknekron Software Systems application that allows traders, brokers, and others to customize the presentation and monitoring of market information. An "object oriented" approach provides a state of the art user interface and display environment; users and system administrators define customized market data information pages or "sheets." A sheet is an arbitrary arrangement of objects, each of which displays an item or group of related items in a pre-defined way. There are standard sheets which come with the product, shared sheets used throughout a department or entire firm, and specialized sheets used by an individual or small group. MarketSheet™ gives the user complete flexibility to organize, format, and display information as the user needs it.

MarketSheet™ receives its market data from the Teknekron Information Bus™ (TIB™) component, a powerful suite of communication protocols that separate information sources, like MarketFeed 2, Ticker III, or Telerate TDPF from information consumers, like MarketSheet™ or Teknekron's RealTime Spreadsheet. This means the user can add market data feeds without obsoleting your applications and the user can add applications without changing feeds.

The MarketSheet™ application is a member of the Teknekron Trader Workstation System family of applications. As such, it employs a number of concepts which are described in the Trader Workstation Introduction manual. Thus it is particularly important to read that manual first.

The following sections are organized as follows: An introduction introduces the features and primary objects of

- 45 -

MarketSheet™. Then a Tutorial explains how to use MarketSheet™, covering many of the basic operations. Finally, a Reference section describes the menus, objects, and scripts in detail. Most of this detail is not needed for a first reading but will be useful after becoming familiar with the program.

This version of the MarketSheet™ program corresponds to Versions 2.1 and 2.2 of the product. These versions differ only in which feeds are supported. Version 2.1 does not support MarketFeed 2 in order to load faster and operate with less memory. No differences are seen by the users.

#### **\*INTRODUCTION**

##### **Features**

Features of the MarketSheet™ application include:

##### **15 • Display**

The display consists of objects on sheets. The user has complete control over the appearance of each object. All objects have characteristics such as foreground color, background color, font, and border which can be changed to suit user requirements and preferences.

##### **• Menus**

MarketSheet™ uses a pull-down menu system. The Menu Bar is always displayed across the top of the window, displaying the names of the available menus. Clicking on a menu title causes the menu to appear below the bar. Selections from these menus are the "commands" that drive MarketSheet™.

##### **• Dialog Boxes**

Pop-up dialog boxes are used to input characteristics associated with each object or sheet, such as market symbol, exchange, graph axis limits, etc. Dialog boxes are also used to specify more detail in many of the commands.

##### **35 • Layout**

- 46 -

There are a number of helpful facilities for arranging items on a sheet, such as user-specifiable grids for alignment, and commands to bring overlapped objects to the front or to the back. Objects can be aligned to the grid either on creation or at any later time. The grid can be adjusted through the sheet edit dialog box, and the grid size is maintained separately for each sheet. Layout commands are consistent among the different tools.

10 • **Scripts**

The user can define a sequence of actions to be carried out when a button is pressed or a price update occurs. These actions are expressed in a macro language called MarketScript. This facility greatly increase the flexibility and generality of the program.

• **Alerts**

The system monitors for alerts (such as upper and lower limits) in each quote or dynamic graph. The alerts can execute scripts in order to change the appearance of monitor items on the screen. The scripts can even run other programs, such as a program to play a sound file for audio alerts, or to log the alert in a file.

• **Display Styles**

Each Quote and Ticker object uses a display style to format its output. These display styles indicate which fields to show (symbol, price, bid, ask, etc.), how wide to make each field, and what colors to use. The flexibility of the style facility allows formats to resemble the Quotron screen displays, or the Equity 2 terminal. The user cannot edit the styles, but a number of standard formats are provided, and the MarketSheet™ Administrator's Manual describes how to edit the styles and add new ones.

• **Files**

35 The user can create multiple MarketSheet™ files that can be saved and reopened, much like the files in the RealTime Spreadsheet. Each file can contain many



- 47 -

sheets.

#### OBJECTS AND TOOLS

The application displays "sheets" or market information. Users can create any number of sheets, which 5 are then organized into an index for each access. Each sheet displays "objects" created by tools in the "toolbox." Available types of objects include:

- **Label**

Labels provide status text to enhance or annotate the 10 display, such as "Market Summary" placed at the top of a sheet.

- **Quote**

Quotes display securities prices and other TIB 15 subjects in numeric format, with prices changing in real-time. Data can be displayed from any 0 elementized feed (such as MarketFeed 2), from shredded page-based services (such as Telerate or Reuters), or from any TIB subject published on the network (such as a calculation published from the 20 RealTime spreadsheet or a C program).

- **Dynamic Graph**

Dynamic graphs chart securities prices and other TIB 25 subjects in real-time. Multiple subjects can be included in a graph and simple arithmetic operations can be performed on the axes, such as spreads. Graphs can also be merged, meaning that two prices can be shown against one time axis.

- **Ticker**

30 Tickers display a list of securities prices and other TIB subjects that scroll as the subjects change in real-time. The user can specify the securities and exchanges to be included in the ticker and set volume thresholds.

- **Page Fragments**

35 Page Fragments are "cut-and-paste" sections from page-based data sources (such as Telerate TDPF or Reuters

- 48 -

RDCDF) and displayed as objects in MarketSheet™. The fragments update in real-time.

- **Buttons**

Buttons are triggers for scripts which are carried out when clicked. These scripts are expressed in the MarketScript™ language. In most cases, these actions are similar to those inaccessible through the menus, but these triggers can be placed on the sheets themselves. For instance, the user can create a button which will bring up a particular sheet when clicked on. This allows the creation of "hypertext links" between related information, such as a security and its options pricing. Available tools include:

- **Grid**

When this tool is active, all creation, movement, and resizing of objects will be adjusted to match the current sheet grid settings.

- **Selection**

This tool is used to select one or more objects for editing, or to move or resize the objects.

- **Duplicate**

This tool is used to make copies of an existing object. The copies are arranged in a grid. This tool is very useful for creating monitor lists or grids.

## 25 TUTORIAL

### Getting Started With MarketSheet™

This section discussed MarketSheet™ informally, introducing concepts in the sequence the user might encounter when experimenting with the program on your own workstation. The Reference section is more structured and covers many of the same topics in more detail.

#### **Basic Display**

The MarketSheet™ display shows a rectangular region, called a Sheet, in the application window, on which any

- 49 -

number of objects may be placed.

#### **Sheet Name**

Figure 1 shows a sample MarketSheet™ display. At the top left of the screen is the Title Bar. Below it is a line 5 containing the Sheet Name indicator, the Index Position indicator and the Menu Bar. The Sheet Name indicator is used to show the name of the Sheet currently being displayed. It can also be used for quick access to sheet attributes. Double-clicking on the Sheet Name indicator 10 opens the Sheet Editor dialog box.

Besides the Name, the attributes of a sheet are the background color, the grid size, how the sheet should be changed when the application is resized, and whether the grid should be shown. These characteristics can be modified 15 by the user.

#### **Index Position**

Next to the Sheet Name indicator is the Index Position indicator. This indicates how many sheets are in the current file and which sheet number is showing. It can also 20 be used for quick access to index information. A double click on the Index Position brings up the Index dialog box, which displays a list of the sheets in your MarketSheet™ file.

#### **Menu Bar**

25 To the right of the Index Position indicator is the Menu Bar shown in Figure 11. The menus pull-down by pressing and holding down on the menu choice using the left Mouse button. The menus are the main way of carrying out commands in the system. Commands provided in the menus 30 support changing and editing the contents of the sheets.

Pressing the left button while you're pointing to a menu title causes the title to be highlighted and the menu to appear. While holding the button down, drag the pointer down through the entries of the menu. As the user drags 35 through a menu, each usable command is highlighted in turn. If the user changes your mind about choosing a command, move the pointer off the menu and release the mouse button.

- 50 -

Nothing is chosen unless the user releases the mouse button while one of the commands is highlighted.

Some menu commands will be shown in gray rather than black. These commands are not available right now. For 5 instance, if there is only one sheet, the command to move to the next sheet is not available. Some menu commands are shown with three dots after their name. These are commands which will bring up a dialog box for further specification of information. For instance, the Create command under the 10 sheet menu brings up a dialog box so that the user can enter the name of the sheet to be created.

Many menu items have a keyboard shortcut which is a way of selecting them without using the mouse. The more commonly used commands have a diamond and a letter to their 15 right. This indicates that the command can also be selected by holding down the diamond key (to the left and right of the space bar), and pressing the specified letter. Note that the title of the menu containing the command will flash when the command is selected, but the menu will not be 20 shown.

### Objects

The sheet in Figure 1 contains the following objects: At the top is a label which displays the text "Market Summary." Below it is a short quote showing the last trade 25 in XYZ. Below that are a graph (not shown in full detail) and a fragment of the Reuters WRLD page. Near the bottom of the sheet are a button and a ticker.

### DIALOG BOXES

Dialog boxes are used in several places where 30 information must be specified about an object or an operation. For instance, double-clicking on an object brings up a dialog box in which its characteristics are shown. A typical dialog box contains the following kinds of items:

- labels and grouping boxes
- 35 • text entry areas
- buttons

- 51 -

- lists

In the example of Figure 12, "Name" is a label, and to its right is a text entry area. To change or enter information in a data entry field, click the left mouse button within the field. A small typing cursor appears at the point of text insertion. Use the Back Space key to delete old text. Type in the next text. The user can also select the text to be replaced by dragging through the characters with the mouse. The characters dragged through will be highlighted, i.e., shown in white on black. The next character typed will replace the selected text.

To select an option via a selection button, click the left mouse button within the square box next to the name of the option. Some options are "toggles" which turn on and off individually each time the user clicks them. Others are "radio buttons," a list of alternatives of which only one will be active at any time.

To use a list, click with the mouse on an entry of the list. These are typically used for lists of color names or sheet names. The Scroll Bar to the right of the list indicates what portion of the list is visible, if not all the entries are showing. Click on the up or down arrows of the scroll bar to see more items. The center part of the scroll bar will indicate what portion of the full list is being shown. Clicking and dragging the mouse on it will enable quick scrolling to the beginning or end of the list.

One item on the dialog box has the Input Focus at any time. This concept is similar to how the Control Panel selects which application window has the input focus. The focus is indicated by drawing a black rectangle around the item. For instance, when a dialog box first appears, the item in the upper left will have the focus. To shift the focus, use the arrow keys to move it left, right, up, or down. In addition, the Tab key can be used to move to the next item, the Shift-Tab key can be used to move to the previous item, and the Keypad Home key can be used to move to the item in the upper left. When a button is traversed

- 52 -

to, it may be "clicked on" by typing the space bar.

When a text entry field is traversed to, its entire contents are highlighted so that the user can easily enter another string. In order to simply edit the existing text, 5 click in the text to specify the insertion point. Alternately, the following keyboard equivalents are available for setting and moving the insertion point through keyboard commands:

	Control-F	moves the insertion point forward
10	Control-B	moves the insertion point backward
	Control-A	moves the insertion point to the beginning
	Control-E	moves the insertion point to the end

To conclude a dialog box operation, click the left 15 mouse button on an action button. These are command names surrounded by a rounded border and, on a color monitor, shaded in color. Most dialog boxes have "OK" or "Cancel" buttons to accept or reject the dialog box action and exit the dialog box. Dialog boxes may also be exited by pressing 20 the Return key which is equivalent to OK, or by pressing the Escape key which is equivalent to Cancel.

#### MANIPULATING SHEETS

A MarketSheet™ file is organized into a list of sheets, or pages, of which only one sheet is shown at a time. Hence, 25 the most common operations are to change which sheet is showing. Operations to show the First, Previous, Next, and Last sheets are provided in the Sheet menu as shown in Figure 13, below. These will navigate through the ordered set of sheets in the system.

30 An alternative way of navigating to other sheets is to use the Page UP and Page Down keys. On the Sun keyboard, these are R9 and R1, corresponding to the placement of the Page Up and Page Down keys on the keypad of an IBM PC.

Yet another way to change sheets is to display the 35 Index dialog box, using the Index operation on the Sheet

- 53 -

menu. This is shown in Figure 14, below. The dialog box contains a list which holds the names of all of the sheets.

To display a sheet, select it and then click on OK, or as a shortcut, double click on its Index entry. The scroll 5 bars may be used if there are more sheets than will fit on the display. The buttons to the left of the list are used to change the ordering of the sheet in the list of sheets.

#### CREATING AND DELETING SHEETS

Besides moving around to the different sheets, the user 10 can also create, delete, rename, and change the order of the sheets. The Sheet Create menu command is used to add a blank sheet. A dialog box will appear in which the sheet name may be entered.

The name of the new sheet defaults to "No Name," and 15 the new sheet will be placed after the current sheet in the index. Note that it is allowable for more than one sheet to have the same name.

To change sheet characteristics such as the name, the background color, or the grid sizes (to be explained below), 20 use the Sheet Edit operation or double-click on the Sheet Name indicator. Complete the dialog entry by clicking on the OK button.

The Sheet Delete menu command is used to delete the current sheet. A dialog box will appear, indicating the 25 name of the current sheet and requiring that the user confirm the deletion. Note that when there is only one sheet, it may not be deleted.

The Sheet Index menu command is used to change the current sheet, as seen previously, and it can also be used 30 to change the order of the sheets. Bring up the dialog box and then click on a sheet name which is not the top one. Clicking on the button marked "Top" will then move that sheet to the top of the list. Click on "OK," and the selected sheet will be moved to the first position. There 35 are also buttons to move a selected sheet forward one position, backward one position, or to the end, and to

- 54 -

create, delete, and rename sheets while the index dialog is active.

## MANIPULATING OBJECTS

### Toolbox

- 5       The Toolbox is a vertically arranged set of icons or buttons along the left side of the display sheet as shown in Figure 1. It is initially not shown. Using the Show Toolbox command in the Edit menu will cause it to appear on the left side of the Display.
- 10       The Toolbox contains an icon to indicate whether the Grid tool is active, then a gap, then an icon to represent the Selection tool (or arrow), then an icon to represent the DuPlication tool, and then an icon for each of the available object types. Only one of the tools can be active at any
- 15 time. It is shown in reverse video. The Grid tool is a toggle, either on or off.

The Toolbox is used to select the next object to be created. After picking a tool (such as a Quote), click and drag to draw a rectangle on the current sheet. The selected

20 type of object will be created, and the Toolbox will reselect the Selection or tool.

If the Grid tool icon is lit, the corners of the drawn rectangle will automatically be adjusted to the grid sizing for the sheet, whether or not the grid is actually being

25 shown. This is a useful page layout feature when creating an array or similar items on a sheet.

## SELECTING OBJECTS

Editing operations provided by the menu system will operate on Selected Object(s) on the current sheet. Selected

30 objects are indicated by a "picture frame corners" around the object, as shown in Figure 16. To select an object, single click on the object with the mouse. A way to select more than one object at a time is to draw a large rectangle (using the Select tool) around the desired objects, as shown

35 in Figure 17.



- 55 -

Holding down the Shift key acts as an "extend selection" feature similar to many programs for the Apple Macintosh, allowing the user to select and highlight more than one object, by clicking on each one of them. If the  
5 user clicks on a selected object, it is unselected.

If the user draws the selection rectangle with the Shift key held down, any objects in the selection rectangle will be toggled, reversing the selection state of all affected objects. The user can unselect all objects by  
10 clicking the left mouse button on an empty area of the sheet.

Another way to select objects is to select all of them, by using the Select All command in the Edit menu.

#### USING KEYBOARD TRAVERSAL

15 The keyboard traversal facility allows the user to change which object is receiving keyboard input, much like the Control Panel function keys are used to change which application receives the keyboard input. This is also the same facility which is used for navigation in dialog boxes.  
20 Not all objects can accept the keyboard traversal; currently only the Quote, Page Fragment, and Button objects do.

The object which has the traversal focus will be shown with highlighting market along the sides, not the corners. This is shown in Figure 18. Note that selection and  
25 traversal are two distinct operations and are indicated by two distinct highlights. When an object is both selected and traversed to, the highlights will merge to form a complete border.

Use the arrow keys to navigate in direction such as up,  
30 down, left and right. The Tab key will move to the next item, and the Shift-Tab will move to the previous item. The tabbing sequence is the same as the stacking order, which defaults to the order in which the items were made. This can be altered using editing operations described below.

35 When there are more than one selected object, one is marked as the Traversal object, meaning that keyboard input

- 56 -

will be sent to it. This facility is mostly used for quick editing of the symbols in Quote objects, or the current page shown in a Page Fragment, which are explained in more detail below.

## 5 DUPLICATING OBJECTS

The Duplicate tool may be used to make copies of objects on a sheet, which will be arranged into a grid of copies having the same size as the original.

To use the Duplicate tool, first create an object, and  
10 set up its characteristics, such as size, colors, font, and other attributes. Place the object in the upper left of the region to be filled with copies. Then click on the Duplicate tool icon in the toolbox. Now click with the left button on the object to be duplicated, and drag right and/or  
15 down to define the region to be filled. As the mouse is dragged, the outlines of the copies to be created will be shown. When the region is complete, release the mouse, and the copies will be made. The Selection tool will be made active again so that the user can operate on the copies.

20 Note that the Duplicate tool cannot be used to make copies of Graph objects, due to their composite structure.

## EDIT MENU OPERATIONS

Once an object is selected, the user can perform editing operations using commands from the Edit Menu.

- 25
- Cut (Delete) the object
  - Front brings the object to the front of the overlap list
  - Back sends the object to the back of the overlap list

30 Since objects can overlap each other, the user may want to bring one object to the foreground, or push one object to the background.

Objects are normally shown, but the user can temporarily hide them, using the Edit Menu.

- 35
- Hide the object

- 57 -

- Show the object

These operations are often used to create pop-up objects on the sheet which are not visible until the user shows them. In most cases, button scripts are used to automatically pop up and down the hidden objects, in order to display help or additional information. This is explained further in the Buttons section.

To make page layout easier, the user can fit objects to the Grid or overlap objects as shown in Figure 19. To see the grid, use the Sheet Edit command to bring up the Sheet Editor Dialog Box. Change the Grid Size by single-clicking anywhere in the Grid Size box. Once the user has created the desired grid size, click on the Draw Gridlines button and close the Dialog Box. The Grid lines will now be drawn on the sheet.

Aligning objects will automatically resize their borders to the size of one grid box, or multiples of the grid box size, if the object is large. When the user has selected the objects to be aligned, choose

- Align

to resize to the grid. Selected objects will align to the grid. Other objects will not move.

In order to refer to specific objects or collections of objects in the MarketScript® language used by the scripts, objects can be given names. Note that the name of a label, for instance, is different from the text shown by the label. Choose

- Name...

to display or change the name. A dialog box will appear, indicating the name of the first selected object. The name can also be changed through this dialog box. The name change will apply to all of the selected objects.

## MOVE AND RESIZE OPERATIONS

Some editing operations do not require the use of the Edit Menu. The user can move or resize an object using the mouse to specify what the user wants to do.

- 58 -

Objects can be Moved by Selecting them, pressing down and holding the left mouse button, and then dragging the objects to a new location on the sheet.

To Resize an object, press and hold the left mouse button on any one of the corners of the Selected object(s). Some degree of precision is required, since the mouse cursor must be within 2 pixels of the corner.

Note that while moving or resizing, the mouse cursor will change into an arrow indicating the movement or direction of resizing.

#### FONTS, COLORS, AND BORDERS

Besides editing a selected object or group of objects, the user can also change their display characteristics. These display characteristics are accessed from the Menu Bar, and include:

- Mode: changes the highlighting mode for page fragment objects. This is described in more detail below.
- Font: changes the font size and style.
- Pen: changes the color for the text and border.
- 20 Border: changes the width for the border of the object.
- Fill: changes the color which fills the background of the object.

Note that the user cannot change the border of Button objects, as their border is drawn internally to provide 25 different shapes.

#### DIALOG BOXES

To edit the detailed characteristics of an object, such as to change a label's text or a quote's subscription information, double click on the object. A dialog box will 30 appear showing that the object's attributes. The dialog boxes are different for the different types of objects, but all have OK, Cancel, and Revert buttons.

For a label, for instance, there is a text entry area for the label's text, and a set of radio buttons to specify 35 the alignment mode of the text.

- 59 -

The specific attributes of the different classes are described more fully in the Reference section of this manual.

#### INTRODUCTION TO SCRIPTS

5       The script facility in MarketSheet® is primarily for advanced users, so only a brief introduction will be given here. Scripts may be used to go to the next page, to a specific page, or to alter the appearance or layout of objects. They are composed of one or more actions, which  
10 are entered as a sentence. For instance, one script would be "Governments find sheet", which would find the sheet named Governments and show it on the screen. This would be faster than bringing up the index dialog box, searching through the list, and selecting the entry for Governments.  
15 Thus, scripts can be thought of as a macro facility for commonly performed functions. Scripts can chain together several operations, such as "next sheet next sheet", which would go to the next sheet, and then the sheet after that.

The section on scripts in the Reference part of this  
20 manual describes the available operations.

#### ALERT FACILITIES

The dialog box for a Quote object allows the user to set upper and lower limits for a specified field. When its limits are reached, a Quote object is shown in reverse  
25 colors (i.e., its pen and fill colors are reversed).

When a Quote reaches a limit value, it will run one more scripts, which the user can enter. There are four scripts associated with each quote, which are run at the start of an alert, the end of an alert, during a normal  
30 update, and during an alert update. This is explained in more detail in the section on Quotes in the Reference Part.

For a Graph, alerts are triggered when the plot line of the instrument moves outside the channel created by an upper or lower trend line.

- 60 -

**LOADING AND SAVING SHEETS AND SHEET SETS**

Only one set of sheets (one file) is active in MarketSheet™ at any time. File menu operations New, Open, Add, Close, and Save are used to create a new sheet set, 5 open (read) an existing one, add another file's sheets to the current file, close the current one, and save the current one. Version shows the Version Number for Market Sheet. Note that while a file is open, the New and Open operations are disabled, and while no file is open, the Save 10 and Close operations are disabled.

The Open and Add operations will bring up a dialog box which lists the files already saved in this account. To select one of these names, simply click on it, or type a new name into the text edit box and MarketSheet® will search the 15 site-wide accounts as well.

Note that the current file name is shown in the application window title bar and will also appear on the icon window title.

The file name that the user types will automatically 20 have the suffix ".ms" added to it. This avoids the possibility that the file will overwrite one of the system files on the disk, which will have a different suffix.

**ON-SCREEN HELP**

There are two ways to obtain on-screen help with 25 MarketSheet™. The user can select Help from the Trader Workstation Teknekron Menu and then select MarketSheet™ from the list of help subjects. Or, the user can click on the Help gadget in the upper right corner of the MarketSheet™ window frame. In either case, a MarketSheet™ Help Menu will 30 appear listing subjects for which help is available. Some of these subjects invoke subsidiary help menus. Once the user selects an item for which help text is available a help window appears on the screen, overlapping other windows. It displays a page of text on the chosen subject. Click the 35 Close gadget in the upper right corner of the help window to erase the window and return the display to its state when

- 61 -

the user requested help.

## REFERENCE

### Menus

Figure 2 shows the basic MarketSheet® display and the 5 menus that the user can select from the main menu bar.

Menu items shown in **bold type** are active in the current product release. The notation "---" appears on the screen as shown in the figure and serves as a logical separator between related groups of menu items. Menus which contain 10 the notation "..." as an item are lists of fonts or colors. What appears on your screen will be a longer, somewhat different list.

Menu items that consist of a command name followed by "..." display a dialog box when selected.

### 15 FILE MENU

Use the File Menu to create new collections of sheets ("files"), to select files, and to save changes the user has made to files. File Menu commands are:

- **New**

20           The File New command creates a new file. Once the user has made a new file, the New and Open commands are disabled until the user closes the file.

- **Open**

25           The File Open command displays a dialog box in which the user enters the name of a previously created file with which the user wishes to work. The dialog box will contain a list of files which are already saved under your account. When the 30           user accepts the dialog box, the file the user named will replace the currently active file. Once the user has opened a new file, the New and Open commands are disabled until the user closes the file.

- 35 • **Add**

- 62 -

5           The File Add command copies the sheets from a file and adds them to the currently displayed file. The additions are temporary until the user saves the current file. Add displays a dialog box into which the user types the name of the source file. The dialog box will contain a list of files which are already saved under your account. Dialog box command Add executes the operation. Cancel exits the dialog box without performing an operation.

10 •       **Close**

          The File Close command removes all sheets from the application, terminating access to the current file. Once the user has closed the file, the Close, Save, and Save As commands are disabled until the user opens a file or create a new file.

15       •       **Save**

          The File Save commands saves the current file. Changes the user has made becomes permanent. The name of the file does not change. Note that if the current file has not been changed, this item is disabled.

20       •       **Save As**

          The File Save As command saves the current file. It displays a dialog box into which the user enters the name of the file. When the user creates a new file the user must save it with Save As in order to assign it an initial name. Later the user can save it again with Save, preserving its name. To create a new file that is similar to an old one, Open the old file, make your changes, and then use Save As to save the new version with a new name.

25       •       **Version**

          The File Version command displays the current software version number in a dialog box. This information is needed when reporting a problem with the software. Click on OK to remove the



- 63 -

dialog box.

#### **EDIT MENU**

Use the Edit Menu to manipulate objects on the current sheet. Edit Menu operations are:

5 • **Undo**

(not implemented in the current release)

• **Cut**

The Edit Cut command deletes all currently selected objects from the sheet.

10 • **Copy**

(not implemented in the current release)

• **Paste**

(not implemented in the current release)

#### **Select All**

15 **Selects all the objects on the current sheet.**

• **Front**

The Edit Front command moves all selected objects to the top of the overlap stack so they become fully visible. Objects that overlap the selected objects will move to the back.

20

• **Back**

The Edit Back command moves all selected objects to the bottom of the overlap stack so they do not obscure any other objects. Objects that overlap the selected objects will move to the front.

25

• **Hide**

The Edit Hide command makes all selected objects invisible on the screen. This command is often combined with the Edit Show command to make objects pop up and down on the sheet.

30

• **Show**

The Edit Show command makes all selected objects visible on the sheet. This command is often combined with the Edit Hide command to make objects pop up and down on the sheet. To select a hidden object for use in this command, use the

35

- 64 -

Select All command, or use the mouse to drag a selection rectangle around the space where the object should be.

- **Align**

5       The Edit Align command adjusts the positions of all selected objects to align with the current grid.

- **Merge**

10       The Edit Merge command merges graphs. Select the price axes of two graphs and then execute the Merge command. The first graph disappears and its price axis is added to the second selected graph, creating a multiple axis graph.

- **Name...**

15       The Edit Name... command is used to display or change the names of the selected objects. This allows the objects to be tagged so that they can be referenced in button scripts. A dialog box will appear, showing the name of the first  
20       selected object, and if a new name is entered, all selected objects will be renamed. Another use of the Name command is to assign a name to a ticker so that its selection list can be copied when defining other tickers.

- 25 • **Show [Hide] Tools**

      The Edit Show Tools command causes the toolbox to appear on the left side of the display area. Use the toolbox to create objects and to control the grid. When the toolbox is displayed, this command becomes Hide Tools. Hide  
30 Tools turn off display of the toolbox.

### **SHEET MENU**

      Use the Sheet Menu to create, delete, and modify sheets and to navigate among existing sheets. Note that it is allowable for more than one sheet to have the same name.  
35 This allows the "find sheet" scripting operation to navigate through a series of sheets.

- 65 -

- **First**

The Sheet First command transfer to the first sheet in the list of sheets in the current file.

- **Prev**

5 The Sheet Prev (Previous) command transfers to the previous sheet in the list of sheets in the current file.

- **Next**

10 The Sheet Next command transfers to the next sheet in an the list of sheets in the current file.

- **Last**

The Sheet Last command transfers to the last page in an the list of sheets in the current file.

- **Index**

15 The Sheet Index command brings up the Index dialog box. This can be used to review the names of the sheets, to select a different sheet, or to change the order of sheets in the file. A short-cut for bringing up the Index dialog box is to double-  
20 click on the index position indicator.

- **Create**

The Sheet Create command brings up a dialog box which allows the user to create a new sheet containing no objects and having the name  
25 "Untitled". Change the name with the Sheet Edit command or double click on the Sheet Name indicator.

- **Delete**

30 The Sheet Delete command brings up a dialog box which allow the user to delete the sheet that is currently displayed. The user must confirm the deletion as a safeguard.

- **Edit**

35 The Sheet Edit command displays a dialog box that lets the user enter a sheet name and background color, turn display of grid lines on and off, and adjust the grid size. A short-cut for bringing up

- 66 -

the Sheet Edit dialog box is to double-click on the sheet name.

#### MODE MENU

Use the Mode Menu to change the highlighting mode in 5 Page Fragment objects. Each page fragment implements the basic highlighting facilities used in Telerate displays to indicate which characters on the source page have changed. The available modes are:

- **None**  
10           No highlighting of changes is performed.
- **Mode J - Numeric Highlighting**  
            The numeric characters which have changed in the most recent update within the region shown by this fragment are shown in reverse colors.
- 15 • **Mode R - Line Highlighting**  
            The line containing the most recent change within the region shown by this fragment is shown in reverse colors.
- **Mode H - Accumulated Numeric Highlighting**  
20           The numeric characters which have changed since the last mode change or highlight clear are shown in reverse colors.
- **Mode F - Accumulated Line Highlighting**  
            The lines which have changed since the most recent  
25           mode change or highlighting clear are shown in reverse colors.
- **Mode Single - used for single fields**  
            The numeric characters which have changed in the most recent update anywhere on the source page are  
30           shown in reverse colors.

In general, use the Modes J through F when showing a large region of the source page. The highlighting will therefore apply to just this region, and will not be affected by updates outside this region. Use the Single 35 Mode when showing a page broken up into a number of small page fragments, such as one for each price. This way the

- 67 -

highlighting will be based on the most recent update anywhere on the source page (if the user had selected mode J for the small fragment, the highlights would remain until there was another update in that same small region).

5 Note that the highlighting is cleared for the selected objects whenever the mode menu is used. For instance, if Mode H is in effect and a number of highlights have accumulated, select Mode H again to clear the highlights and preserve the mode.

10 Clearing of highlights can also be accomplished through a script, see the Scripts section for more detail.

#### **Font Menu**

Use the Font Menu to change the font used in selected objects. The default font is a small fixed-width font. The  
15 elements of the font menu are defined by your system administrator, and may be changed when other fonts become available. Not all fonts are provided on all X Window System implementations. Therefore, the program determines which of the requested fonts are not available, and  
20 indicates those in gray (disabled) items. The current font is indicated by a black dot next to its name.

Note that if the selected object was made on another system which had a different set of fonts in the font menu, there may be no current font indicated on the menu.

#### **25 Pen Menu**

Use the Pen Menu to specify the foregoing color of selected items. The default color is black. This is the color of the text or lines in the object. The current foreground color is indicated by a black dot next to its  
30 name.

#### **Border Menu**

Use the Border Menu to specify the size of the border around selected items. The default is one pixel. Wider borders waste some screen space but they are more attractive  
35 and easier to manipulate with the mouse. The current border size is indicated by a black dot next to its name. The lower portion of this menu specifies the border color of the

- 68 -

object. The current border color is indicated with a black dot next to its name.

Note that the border color and width of buttons cannot be changed.

#### 5 Fill Menu

Use the Fill Menu to specify the background color of selected items. The default color is white. The current background color is indicated by a black dot next to its name.

#### 10 Note on Use of Variable-Width Fonts

Font are divided into two types: fixed and variable. Fixed width fonts have all characters on the same size, while variable width fonts are spaced proportionally. Using variable width fonts on Quotes, Tickers, or Page Fragments will have some unusual effects, because each of those objects are organized around a row-column grid, or matrix, or characters.

The character grid layout facility must allocate space for a grid of characters big enough to hold the biggest character in the font, such as the letter 'W'. Thus it will often make itself wider than it appears to need to be. Likewise, if the width of one of those types of objects is reduced, the character grid layout facility will compute how many columns of the biggest character will fit, and show only that number of character. This can often be deceptive.

It is preferable to use only fixed-width fonts on these types of objects. Not only will this avoid the above effect, but multi-line page fragments and quotes will continue to have their character columns lined up, as on the source page. Since fonts vary from system to system, contact the System Administrator for information on which fonts are variable-width.

#### HOW TO BUILD THE MarketSheet® PROGRAM

##### OVERVIEW

There are two stages. The first stage builds the

- 69 -

Widget get, called Strata, which is used by the application. This stage relies only on availability of the standard X11R3 distribution of the X Window System API library and the X Window System X Toolkit API library, both of which are distributed by the MIT X Consortium, and the TIB API libraries.

The second stage is to build MarketSheet™ itself, which uses the Strata widget get library, the TIB API libraries, and the X Window System libraries.

#### 10 FIRST PHASE

The following files are used to build the Strata library:

- Axis.c
- Axis.h
- AxisP.h
- 15 Basic.h
- BasicP.h
- Box.c
- Box.h
- BoxP.h
- 20 Button.c
- Button.h
- ButtonP.h
- CharGrid.c
- CharGrid.h
- 25 CharGridP.h
- Converters.c
- Converters.h
- DBoard.c
- DBoard.h
- 30 DBoardP.h
- Dialog.c
- Dialog.h
- DialogP.h
- Gadget.c
- 35 Gadget.h
- GadgetP.h
- GraphData.c
- GraphData.h
- GraphDataP.h
- 40 GraphMgr.c
- GraphMgr.h

- 70 -

```

    GraphMgrP.h
    GraphView.c
    GraphView.h
    GraphViewP.h
5 Highlight.c
    Label.c
    Label.h
    LabelP.h
    List.c
10 List.h
    ListP.h
    Manager.c
    Menu.c
    Menu.h
15 MenuBar.c
    MenuBar.h
    MenuBarP.h
    MenuP.h
    Primitive.c
20 Reader.c
    Reader.h
    Selection.c
    TextEdit.c
    TextEdit.h
25 TextEditP.h
    Traversal.c
    Writer.c
    Writer.h

```

Makefile for First Phage (using GNUmake program):

```

30 lib_name = libstrate.a
   lib_members =      Primitive.o Manager.o Traversal.o Highlight.o
                        Selection.o
   \
                        Converters.o Reader.o Writer.o Gadget.o Label.o
35 Button.o \
                        Box.o DBoard.o Dialog.o MenuBar.o Menu.o List.o
   TextEdit.o \
                        CharGrid.o Axis.o GraphView.o GraphData.o GraphMgr.o
   S(lib_name) :      S(lib_members)

```

#### 40 SECOND PHASE

The following files are used to build the MarketSheet™ application:  
 Box.c



- 71 -

Box.h  
BoxP.h  
Button.c  
Button.h  
5 ButtonP.h  
CharGrid.c  
CharGrid.h  
CharGridP.h  
DSAxis.c  
10 DSAxis.h  
DSAxisP.h  
DSGraphData.c  
DSGraphData.h  
DSGraphDataP.h  
15 DSGraphView.c  
DSGraphView.h  
DSGraphViewP.h  
Fragment.c  
Fragment.h  
20 FragmentP.h  
Manager.c  
PlaneMgr.c  
PlaneMgr.h  
PlaneMgrP.h  
25 Publisher.c  
Publisher.h  
PublisherP.h  
Quote.c  
Quote.h  
30 QuoteP.h  
Reader.c  
Subscription.c  
Subscription.h  
SubscriptionP.h  
35 TBAxis.c  
TBAxis.h  
TBAxisP.h  
TBGraphData.c  
TBGraphData.h  
40 TBGraphDataP.h  
TBGraphView.c  
TBGraphView.h  
TBGraphViewP.h  
Table.c  
45 Table.h

TableP.h  
Ticker.c  
Ticker.h  
TickerP.h  
5 TimeGrid.c  
TimeGrid.h  
TimeGridP.h  
bits.arrow  
bits.button  
10 bits.clone  
bits.dsgraph  
bits.fragment  
bits.global  
bits.grid  
15 bits.label  
bits.publisher  
bits.quote  
bits.table  
bits.tbgraph  
20 bits.ticker  
bricks.bits  
button.c  
dsgraph.c  
files.c  
25 fragment.c  
global.c  
items.c  
label.c  
menus.c  
30 mondrian.bits  
meney.bits  
ms.h  
ms23.c  
msDefaults.cf  
35 msEmpty.cf  
msNTib.cf  
page.h  
pagehandler.c  
pagemap.c  
40 pagemap.h  
publisher.c  
quote.c  
script.c  
sheets.c  
45 stylemap.c

- 73 -

```

stylemap.h
table.c
tbqgraph.c
tib.c
5 tib.h
ticker.c
time.c

```

Makefile for Second Phase (using GNUmake program):

```

objects = Reader.o Manager.o PlaneMgr.o TimeGrid.o CharGrid.o Box.o
10 Button.o\
tib.o time.o menus.o sheets.o items.o tools.o files.o \
script.o stylemap.o label.o Subscription.o Quote.o quote.o
\
Ticker.o ticker.o pagemap.o pagehandler.o \
15 Fragment.o fragment.o \
TBXig.o TBGraphView.o TBGraphData.o tbgraph.o DSXig.o \
DSGraphView.o DSGraphData.o dsgraph.o Publisher.o
publisher.o \
Table.o table.o button.o global.o
20 libs = -lnrti -lrti -ltibinfo -lmdp new -lmdsg -lmsa -lsags -lforms
lchan \
-leventX -lciutil -lstrate -ltss -lutil -lXt -lX11
ms23: $(objects) $(libs)

```

Although the invention has been described in terms of  
25 the preferred and several alternative embodiments described  
herein, those skilled in the art will appreciate other  
embodiments and modifications which can be made without  
departing from the spirit and scope of the teachings of the  
invention. All such modifications are intended to be  
30 included within the scope of the claims appended hereto.

- 74 -

What is claimed is:

1. An apparatus for creation of user defined video displays showing the values of real time data generated by a complex system, comprising:

5 a computer means for storing and executing one or more programs;

network means for coupling said computer means to a system being monitored by receiving real time data and for transmitting data to other parts of said system;

10 a display coupled to said computer means for displaying the results of processing by said programs; and

wherein said one or more programs include composition means for accepting commands from a user to create instances of different classes of active objects which display on said display either real-time, changing data generated by said system being monitored or alphanumeric data entered by the user in a format and style selected by said user, and for generating one or more active objects for storing real-time data and/or data entered by said user and associated style maps defining which of said data is to be displayed, and where and how said data is to be displayed, and having associated display objects for storing internal computer data representations of the data to be displayed in accordance with said style map, and including means to request and receive real-time data requested by said user and to update the internal computer data representations of the data to be displayed with new values for the real time data selected by said user for display.

2. In a computing environment including a system generating real time data to be monitored, at least one computer coupled to said system by a communication path and

- 75 -

communication software to monitor said system via data displayed on a display, and one or more programs for controlling the processing of at least one said computer, the improvement comprising one or more programs for creating  
5 a dynamically changeable document on said display having a user defined layout and having user defined display objects displayed at user defined locations, said one or more programs comprising:

label means for providing a text layout tool  
10 accessible by a user such that one or more user defined areas of user defined text may be established in the user defined layout for said dynamically changeable document; and

real time data means for providing a layout tool  
15 accessible by a user such that one or more user defined areas of said dynamically changeable document may be designated by said user to always show the current and dynamically changeable value of any one or more user defined items of real time data.

20 3. The apparatus of claim 2 further comprising graph means for providing a layout tool accessible by a user such that one or more user designated areas of said dynamically changeable document may be defined to always show the dynamically changeable value of one or more user defined  
25 items of real time data as a graph showing the history and current value of the changing value of said user defined items of real time data versus time.

4. The apparatus of claim 2 further comprising ticker means for providing a layout tool accessible by a user such  
30 that one or more user designated areas of said dynamically changeable document may be defined to always show the dynamically changeable value of one or more user defined items of real time data as the current price for each issue for all trades in a user specified set of financial  
35 securities.

- 76 -

5. The apparatus of claim 2 further comprising page fragment means for providing a layout tool accessible by a user such that one or more user designated areas of said dynamically changeable document may be defined to always  
5 show the dynamically changeable value of one or more user defined items of real time data in the form of user designated portions of one or more pages of financial data from a user defined, page oriented financial information service which publishes said financial data via said  
10 communication path.

6. The apparatus of claim 2 further comprising page means for providing a layout tool whereby a plurality of pages may be defined, each having a user defined layout of user defined text and each capable of having user defined  
15 items of real time data displayed in user designated locations thereon.

7. The apparatus of claim 2 further comprising:  
alarm means for providing a tool which may be invoked by the user to establish alarm limits which are  
20 associated with one or more of said user designated items of said real time data, said alarm limits being compared with the value of said corresponding item of real time data from time to time and for setting an alert status when any one of said alarm limits is  
25 exceeded; and

script means coupled to each said alarm limit for storing and executing user defined sequences of commands to cause a user defined sequence of events to occur whenever the corresponding alarm limit is  
30 exceeded.

8. The apparatus of claim 3 further comprising alarm means coupled to said graph means for providing a tool whereby a user may define up to two active trend lines, typically upper and lower limits, for the value depicted on

- 77 -

each said user defined graph, and for comparing each new graph point displayed upon receipt of an update to the corresponding real time data to any active user defined trend lines for the corresponding graph, and if a limit 5 established by an active trend line is equalled or exceeded, for generating an alert signal.

9. The apparatus of claim 2 wherein said real time data means includes quote means for providing a tool which may be invoked by a user to locate at user defined locations 10 on said dynamically changeable document one or more quote active objects, each said quote active object for receiving and displaying the current price for a user designated financial issue at said designated location of said dynamically changeable document.

15 10. The apparatus of claim 2 further comprising:  
button means for providing a tool which may be invoked by the user to locate one or more button active objects at user definable locations on said dynamically changeable document, and  
20 further comprising script means associated with each said button means for storing and executing a user defined sequence of commands which cause a user defined sequence of events to occur, each said script means coupled to a corresponding button active object such  
25 that when a button active object is selected in any manner, said associated script means is invoked and the corresponding script of commands is executed.

11. The apparatus of claim 2 wherein said real time data means includes means to create an active object in 30 memory of at least one said computer for each said user designated item of real time data, each said active object having an associated list of properties and operations which can be invoked by a user to change the values of various properties of the object, wherein each operation can be

- 78 -

invoked by user entry of a command to identify said active object and start the operation and pass an argument to the operation for use in changing the properties of said active object.

5        12. The apparatus of claim 11 wherein each said active object is represented in memory by a record containing the values for each said property and a pointer to a library of programs which carry out the operations associated with that active object.

10       13. The apparatus of claim 12 wherein each said active object comprises:

15           a data object means for storing the internal computer representation of a set of data items about the subject associated with that particular data object;

          a style map means for storing user defined style data specifying particular ones of the data items stored in said data object to display and where and how to display these data items;

20           screen rendering means for rendering information on said display;

          a display object means for extracting the data specified by said style map means from said data object means and for sending said extracted data an  
25       appropriate rendering command based upon said style data to said screen rendering means for rendering said data on said display in accordance with said style data;

30           event trigger means for storing user defined event trigger data specifying the conditions under which the user desires further action to occur; and

          event script means for storing a user defined script of actions which are to be carried out in a user defined sequence if one of the event trigger conditions  
35       is met.



- 79 -

14. The apparatus of claim 13 wherein said display object means includes style processor means for receiving data updates comprised of the new value for one or more items of said real time data associated with one or more  
5 active objects displayed on said dynamically changeable document, and for extracting the corresponding data items whose values have been updated from said data object means, and for updating the extracted data with its new values and storing the updated values in said display object means in  
10 accordance with said style data stored in said style map means, and for comparing the updated values for said data to any event trigger data stored in said event trigger means, and for computing special effects to be displayed on said display according to said style data if any event trigger  
15 has occurred, and for updating the data stored in said display object means if any special effects have been triggered by said update, and for sending appropriate data to said screen rendering means to render any triggered special effects.

20 15. The apparatus of claim 3 wherein said graph means includes means for scaling each axis separately.

16. The apparatus of claim 3 wherein said graph means includes means for displaying the numeric values on each axis of said graph for any particular point on the graph for  
25 any particular point on the graph which has been graphically selected.

17. The apparatus of claim 3 wherein said graph means includes means for causing automatic rescaling of the time  
30 axis for any graph wherein one of the axes has been user defined to show time.

18. The apparatus of claim 3 wherein said graph means includes means for providing a tool whereby a user may graphically designate any two points on a graph so as to

- 80 -

designate a trend line.

19. The apparatus of claim 3 wherein said graph means includes zoom means for allowing a user to enlarge the view of any user designated portion of any said graph.

5        20. The apparatus of claim 3 wherein said graph means includes merge means for allowing a user to designate at least two items of said real time data to have their values graphed on the same graph against the same time axis or other measure.

10       21. The apparatus of claim 7 wherein said script means comprises:

         means for selection of active objects;

         means for editing selected active objects;

15       means for moving and resizing of selected active objects;

         means for moving between pages of said dynamically changeable document;

         means for editing a selected page of said dynamically changeable document;

20       means for causing audible signals, for causing a command of another program such as an operating system program to be executed and for pausing a user designated amount of time; and

25       means for controlling the display of the computing environment such as displaying or not displaying the available tools for creating and editing said pages of said dynamically changeable document or displaying or not displaying alignment aids for layout of said pages of said dynamically changeable document such as grid  
30       lines.

22. The apparatus of claim 1 or 2 or 3 or 4 or 5 or 6 or 7 or 8 or 9 or 10 or 13 or 14 or 15 or 16 or 17 or 18 or 19 or 20 or 21 further comprising publishing means for

- 81 -

providing a tool whereby a user can designate any portion of any page of said dynamically changeable document to be published via said communication software and said communication path to other processes running on any said  
5 computer.

23. The apparatus of claim 10 wherein said script means comprises:

- means for selection of active objects;
- means for editing selected active objects;
- 10 means for moving and resizing of selected active objects;
- means for moving between pages of said dynamically changeable document;
- means for editing a selected page of said  
15 dynamically changeable document;
- means for causing audible signals, for causing a command of another program such as an operating system program to be executed and for pausing a user designated amount of time; and
- 20 means for controlling the display of the computing environment such as displaying or not displaying the available tools for creating and editing said pages of said dynamically changeable document or displaying or not displaying alignment aids for layout of said pages  
25 of said dynamically changeable document such as grid lines.

24. The apparatus of claim 2 further comprising table tool means for providing a tool to create user defined table tool objects for display of useful reference information  
30 such as the names of commonly used pages of page oriented services or securities, etc.

25. The apparatus of claim 21 or 23 wherein said script means further comprises glossary means for providing a tool whereby a user may enter new commands which are

- 82 -

available for execution during a user defined script of actions.

26. The apparatus of claim 2 further comprising means for displaying a menu of universal commands which can be  
5 used to edit or manipulate all pages and all active objects displayed on any page of said dynamically changeable document, and to display a pop-up menu of further commands associated with any command from said menu of universal commands.

10 27. The apparatus of claim 2 or 3 or 4 or 5 or 9 or 10 or 26 further comprising means for displaying all tool means as icons which may be graphically selected.

28. The apparatus of claim 2 further comprising means for highlighting the display of data that has changed.

15 29. The apparatus of claim 11 further comprising means for displaying said active objects in layers, and further comprising means for changing the order of the layers.

30. The apparatus of claim 2 wherein each said page of said dynamically changeable document is simultaneously  
20 displayed in its own separate window.

31. The apparatus of claim 2 wherein said communication software implements a subscription paradigm such that application programs linked thereto need only send a subscription request on a subject to said communication  
25 software and said communication software then carries out appropriate processing to locate a source for the desired information, communicate with that source and enter a subscription such that real time data is constantly sent via said communication path to said communication software and  
30 passed therefrom to said real time data means where the updated values thereof are displayed as they arrive, and

- 83 -

wherein said real time data means includes means for passing subscription requests on user designated subjects represented by the real time data associated with each said active object to said communication software as a  
5 subscription request so as to invoke the appropriate processing to cause real time data to start flowing to update the data of each said active object and the displayed value thereof.

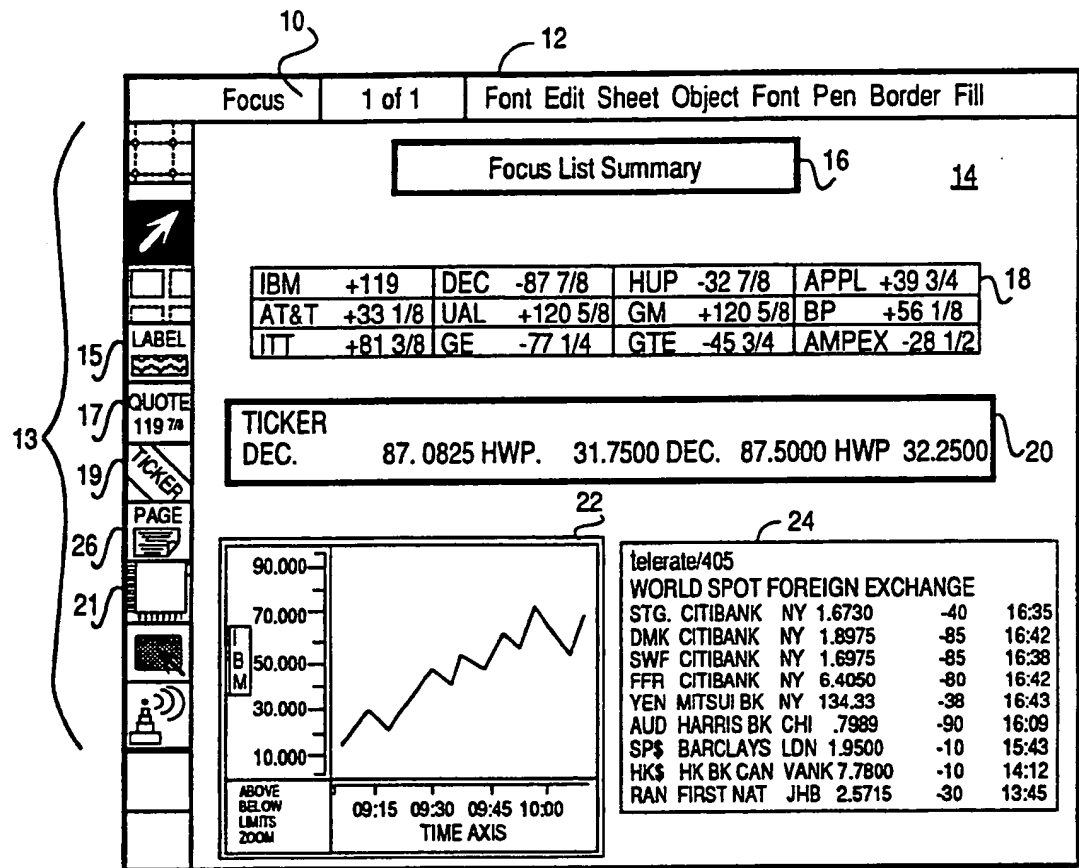


FIG. 1

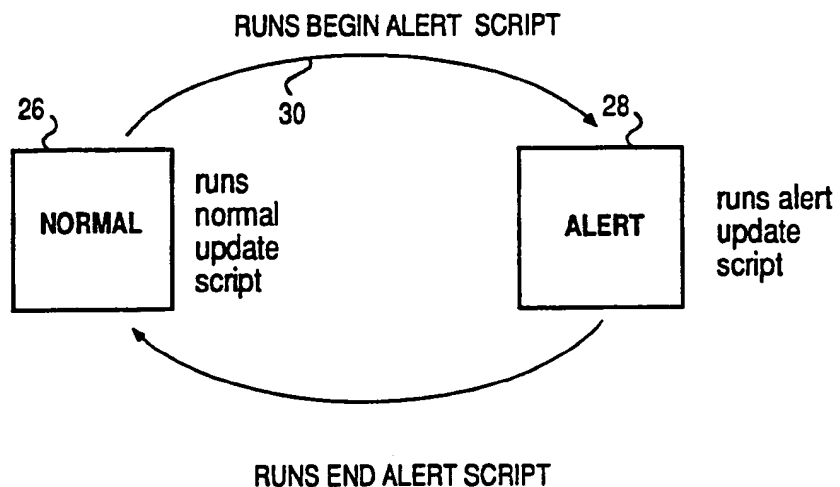


FIG. 2

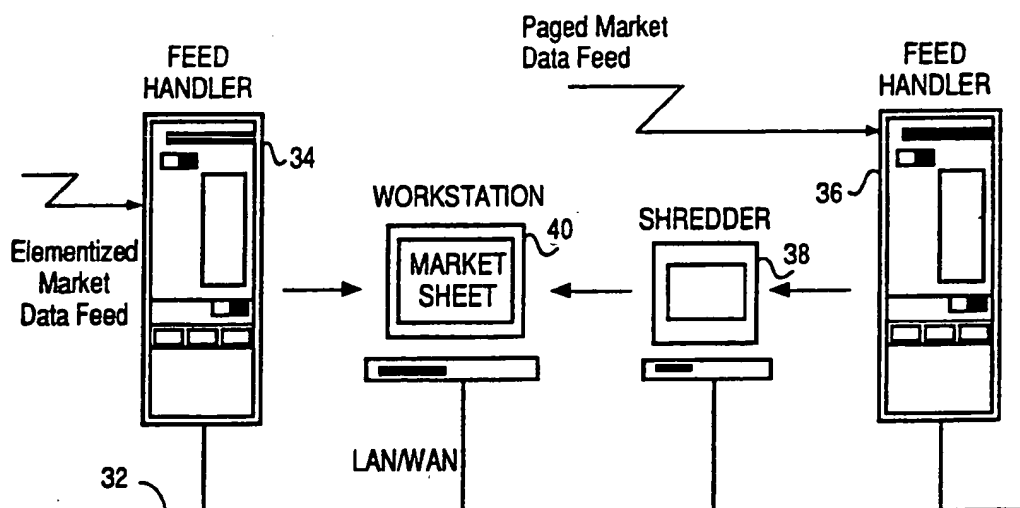


FIG. 3

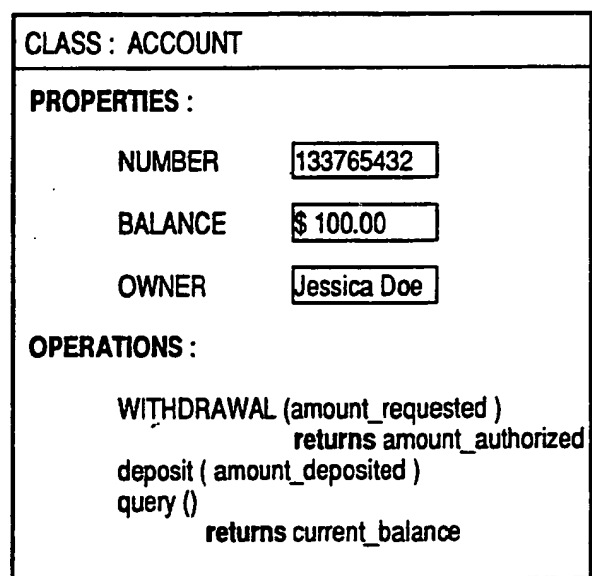


FIG. 4

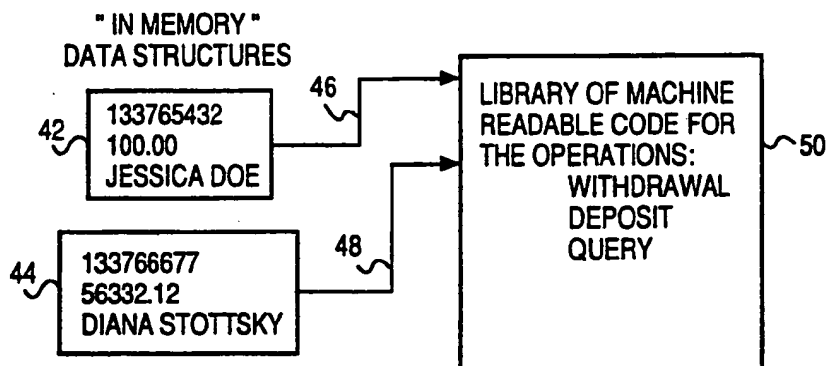


FIG. 5

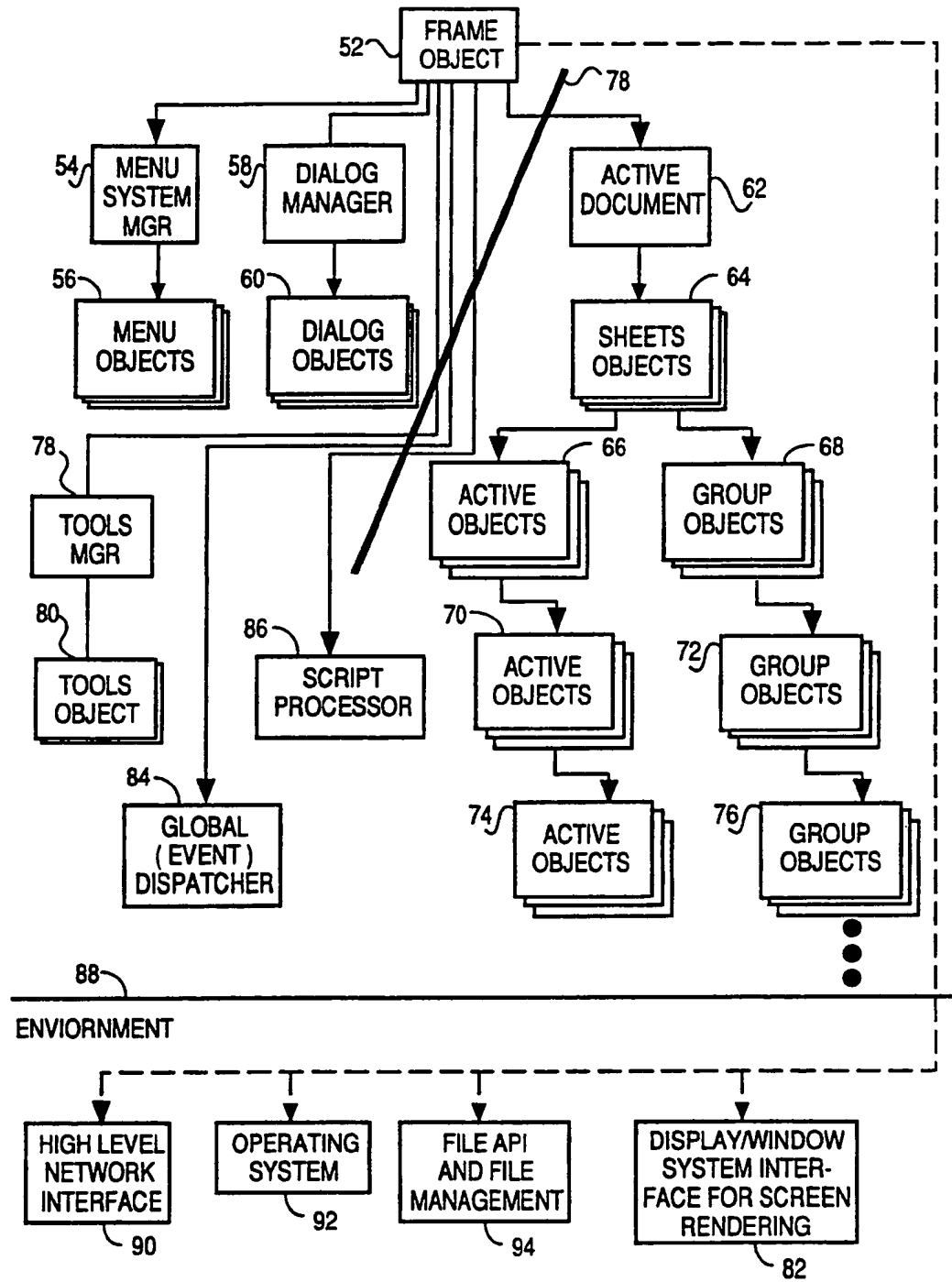


FIG. 6



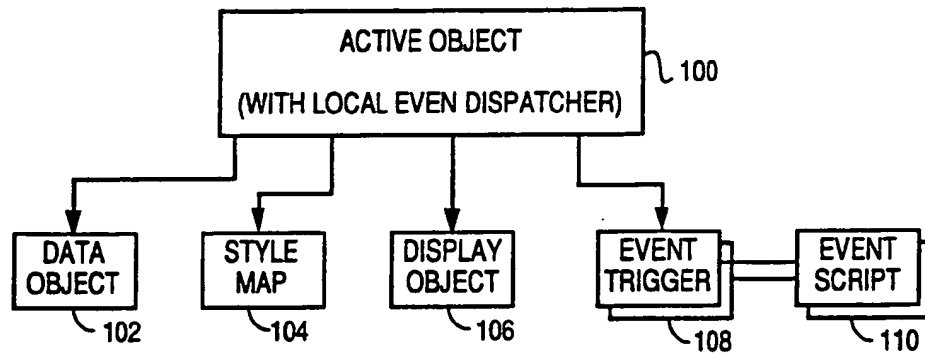


FIG. 7

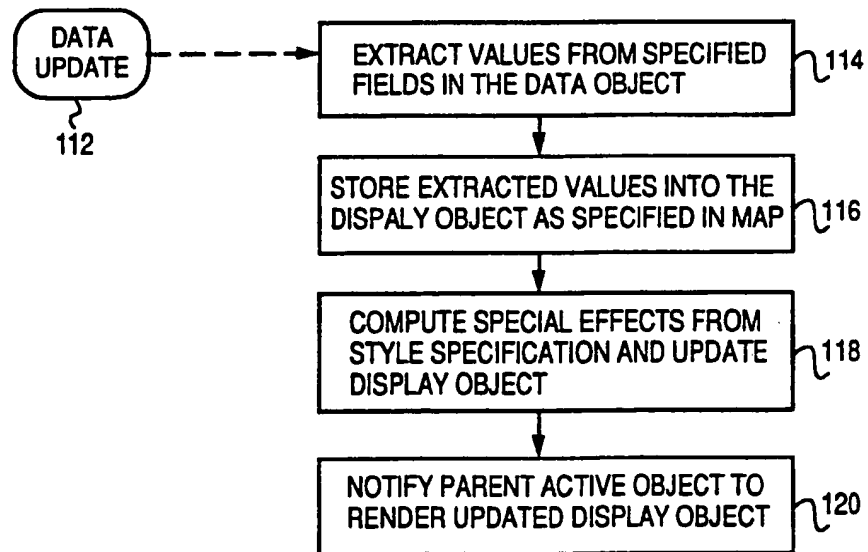
**ACTIVATION  
EVENT**

FIG. 8

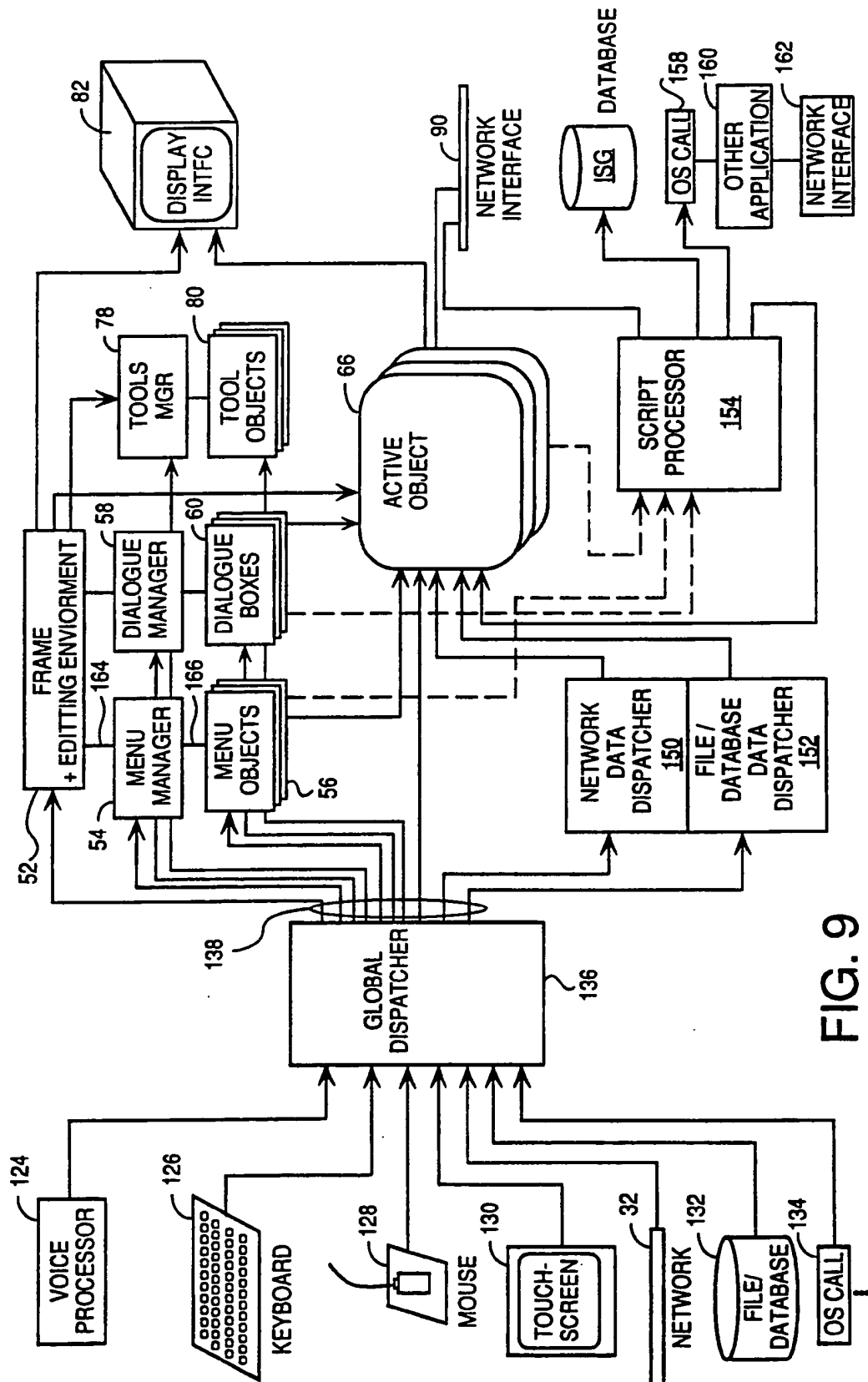


FIG. 9

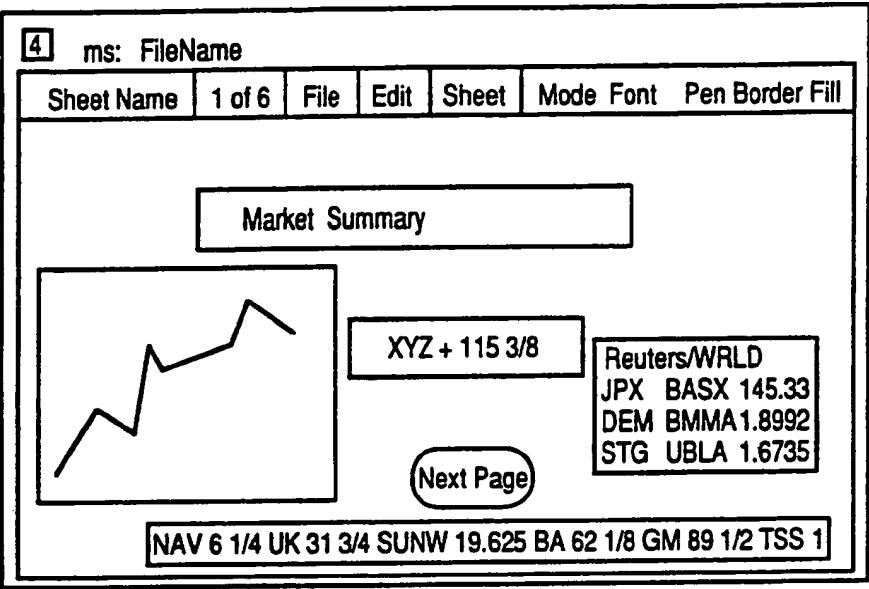


FIG. 10



FIG. 11

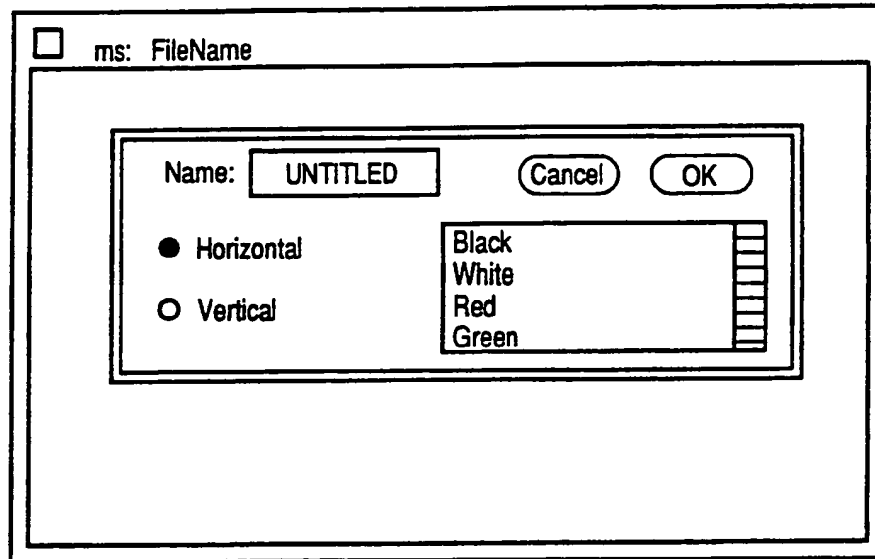


FIG. 12

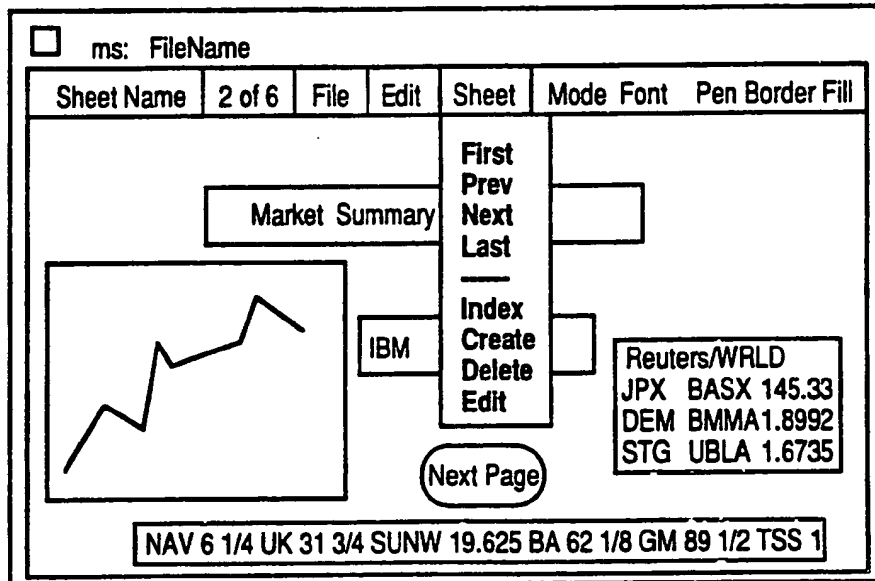


FIG. 13

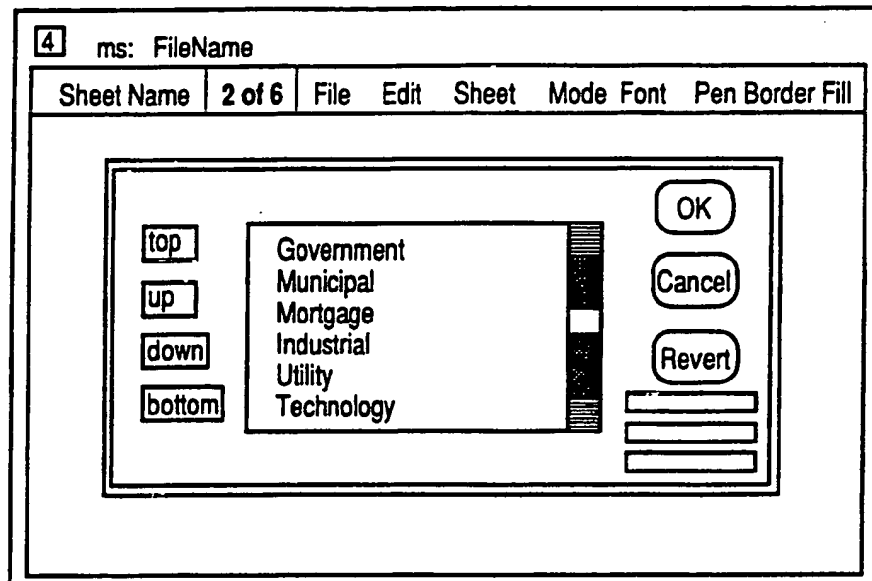


FIG. 14

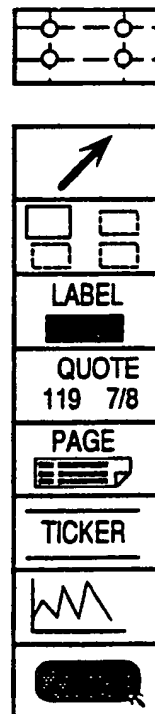


FIG. 15

4/10

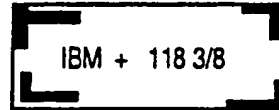


FIG. 16

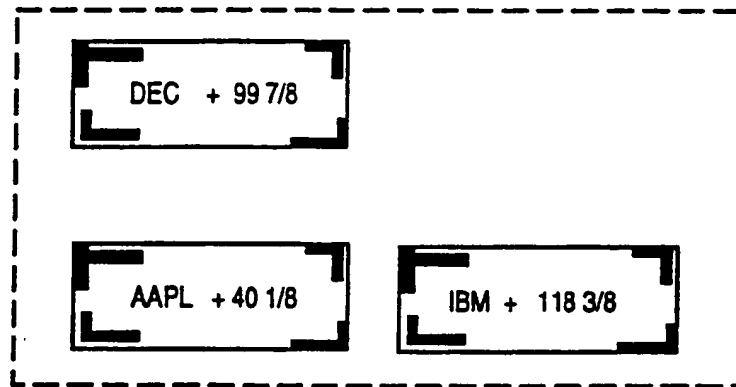


FIG. 17

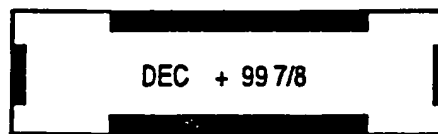


FIG. 18



FIG. 19

10/10

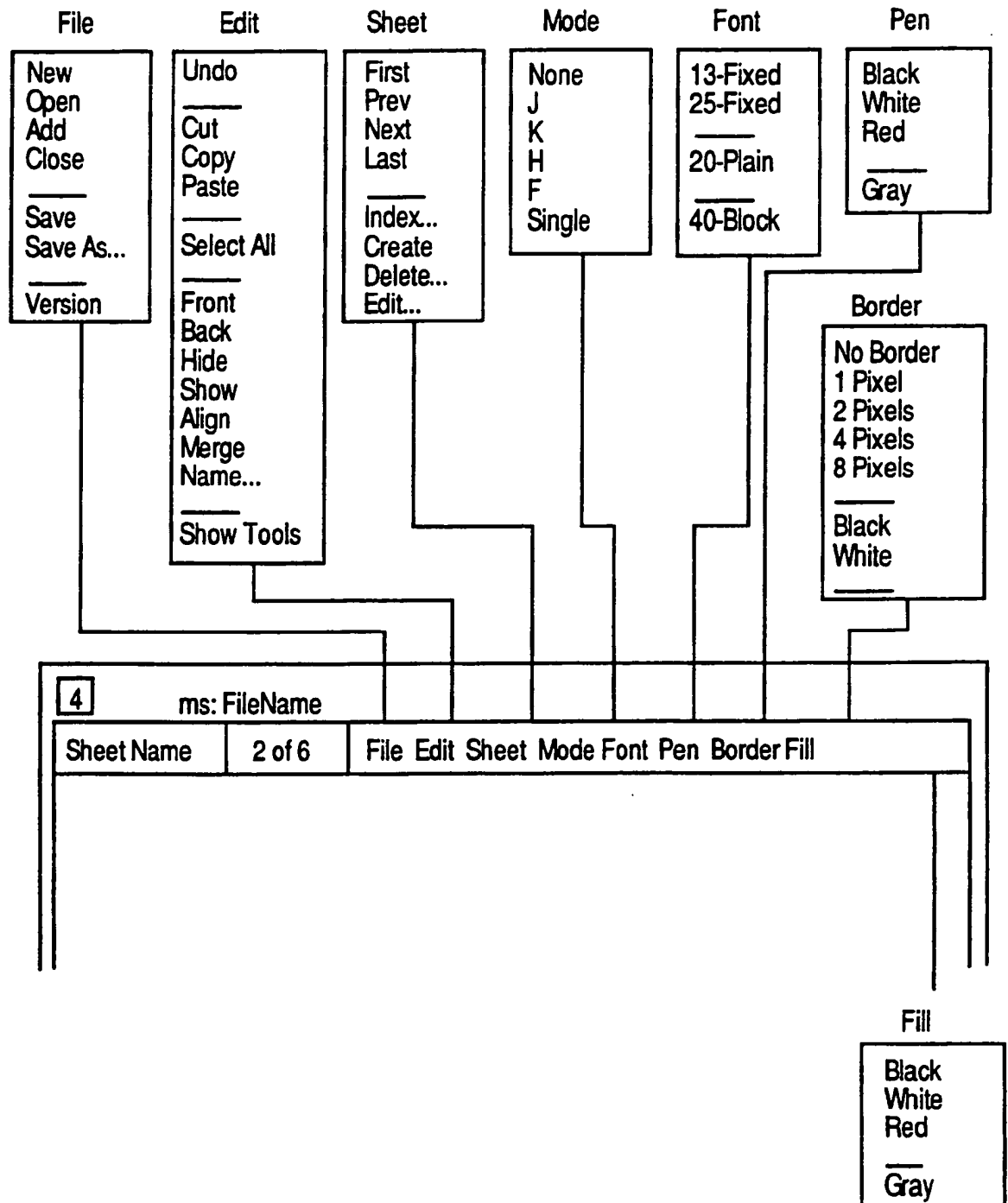


FIG. 20

SUBSTITUTE SHEET

# INTERNATIONAL SEARCH REPORT

International Application No. PCT/US91/09811

<b>I. CLASSIFICATION OF SUBJECT MATTER</b> (if several classification symbols apply, indicate all) <sup>6</sup>	
According to International Patent Classification (IPC) or to both National Classification and IPC	
IPC(5): G06F 15/20	
US CL : 395/155,159	
<b>II. FIELDS SEARCHED</b>	
Minimum Documentation Searched <sup>7</sup>	
Classification System	Classification Symbols
US CL	395/148,153-159
INT CL	G06F 15/20
Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched <sup>8</sup>	

<b>III. DOCUMENTS CONSIDERED TO BE RELEVANT <sup>9</sup></b>		
Category <sup>10</sup>	Citation of Document, <sup>11</sup> with indication, where appropriate, of the relevant passages <sup>12</sup>	Relevant to Claim No. <sup>13</sup>
A	US, A, 4,642,790 (MINSHULL ET AL.) 10 FEBRUARY 1987 See the entire document.	1-31
Y	US, A, 4,750,135 (BOILEN) 07 JUNE 1988 See columns 5-12.	1-31
Y	US, A, 4,845,645 (MATIN ET AL.) 04 JULY 1989 See the entire document.	1-31
A	US, A, 4,943,866 (BARKER ET AL.) 24 JULY 1990 See the entire document.	1-31
Y,P	US, A, 5,058,185 (MORRIS ET AL.) 15 OCTOBER 1991 See columns 6-10.	1-31

- <sup>14</sup> Special categories of cited documents:

  - "A" document defining the general state of the art which is not considered to be of particular relevance
  - "E" earlier document but published on or after the international filing date
  - "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  - "O" document referring to an oral disclosure, use, exhibition or other means
  - "P" document published prior to the international filing date but later than the priority date claimed

- <sup>15</sup> later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
  - <sup>16</sup> document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step
  - <sup>17</sup> document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
  - <sup>18</sup> document member of the same patent family

<b>IV. CERTIFICATION</b>	
Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report
06 FEBRUARY 1992	<b>19 MAR 1992</b>
International Searching Authority	Signature of Authorized Officer
ISA/US	HEATHER HERNDON



1/10

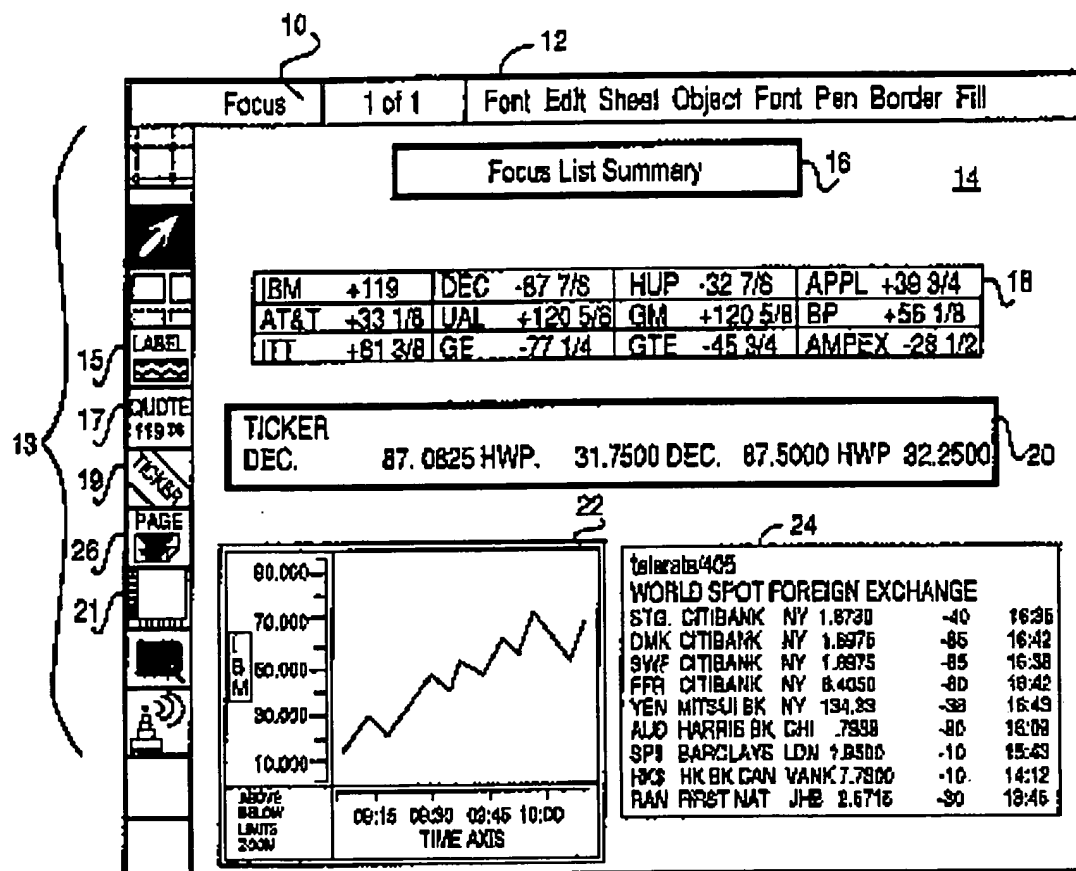


FIG. 1

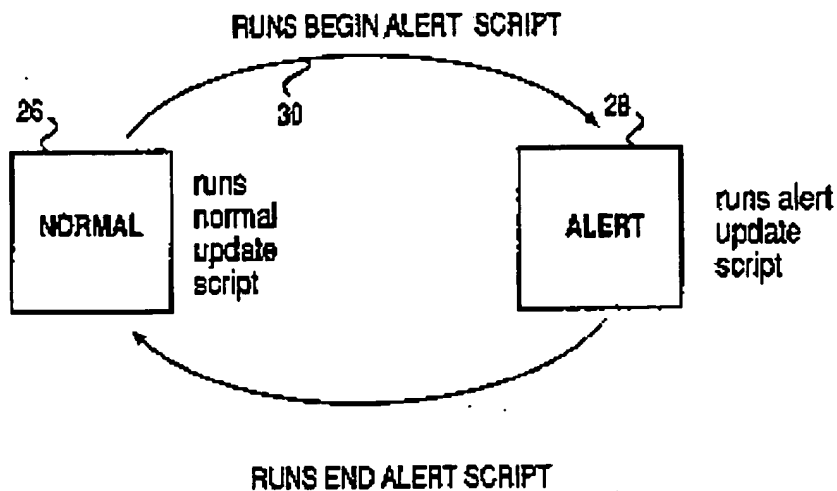


FIG. 2

2/10

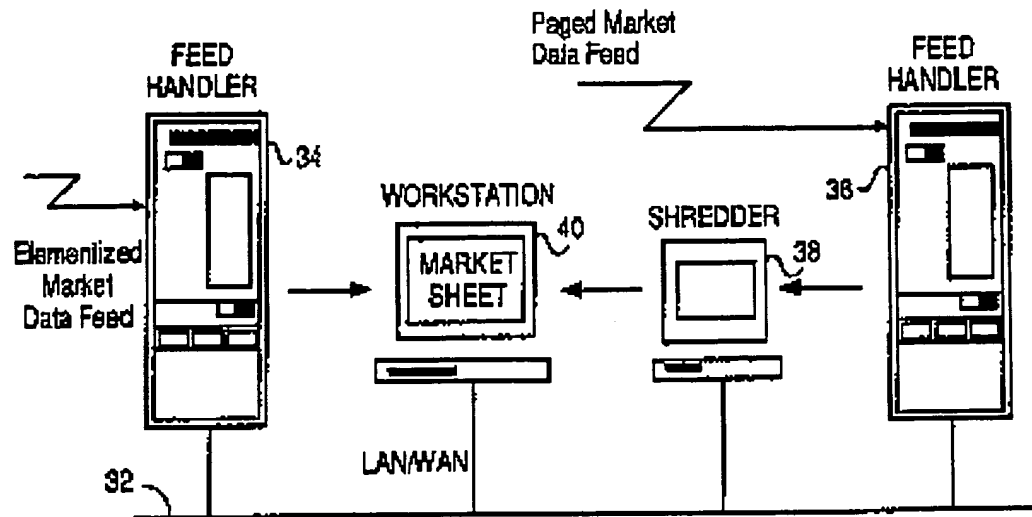


FIG. 3

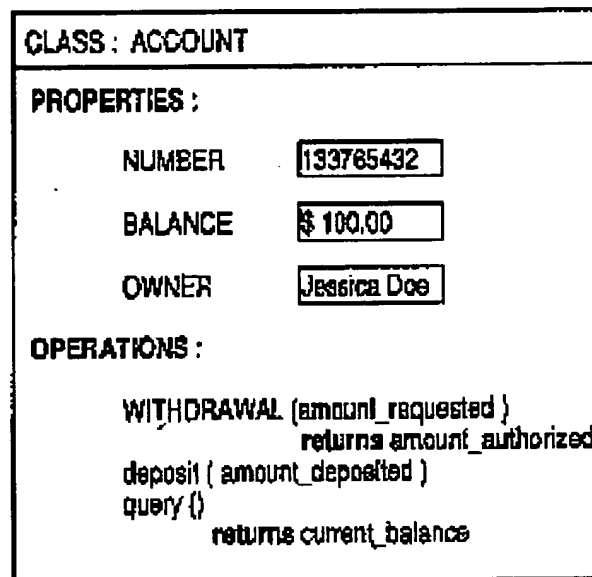


FIG. 4

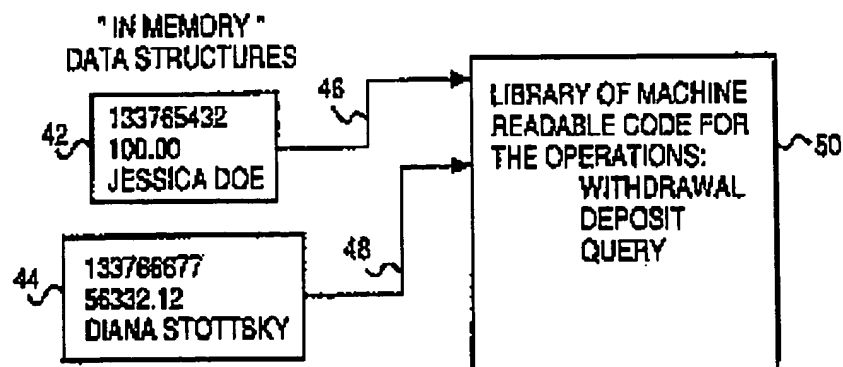


FIG. 5

3/10

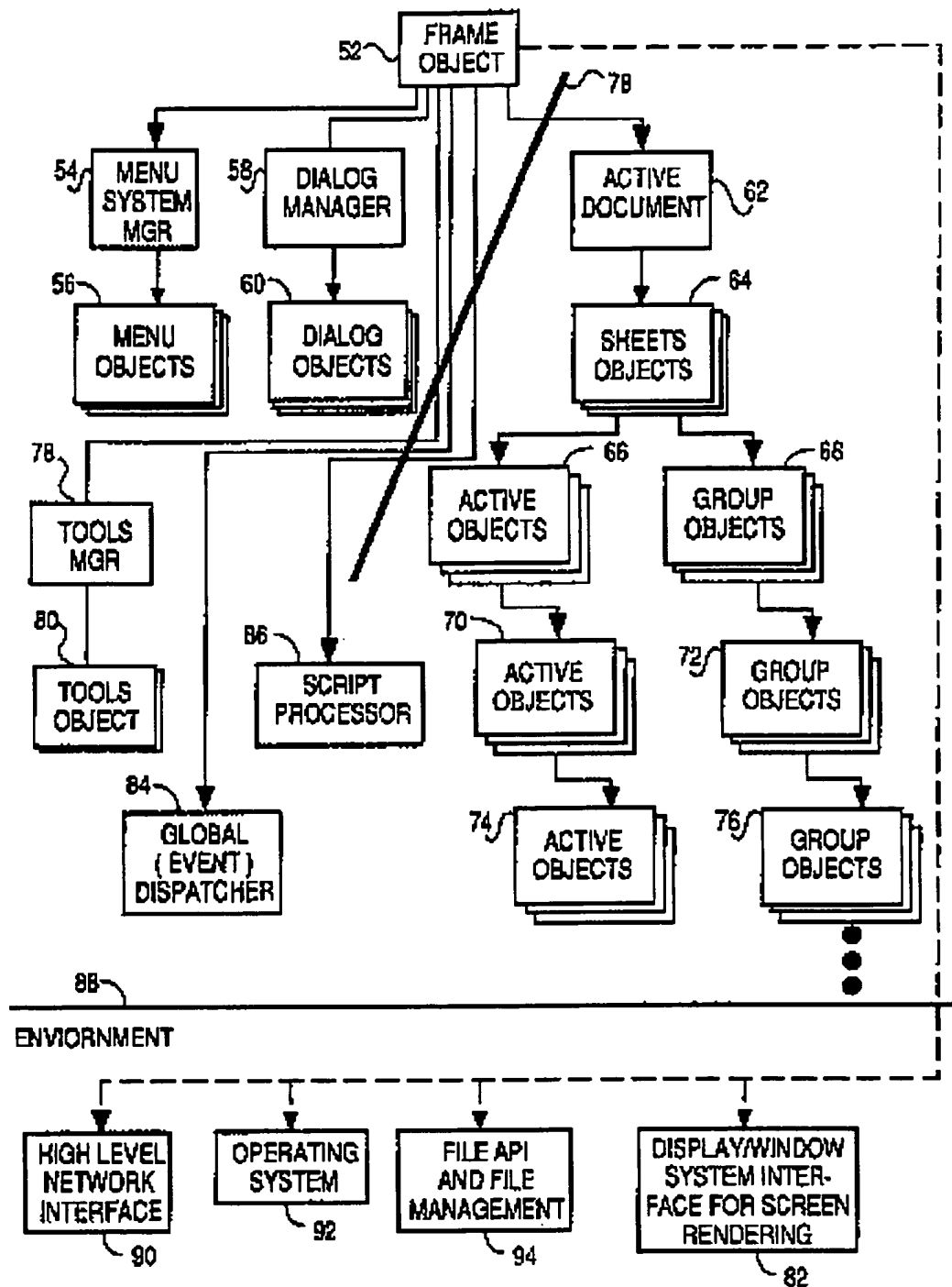


FIG. 6

4/10

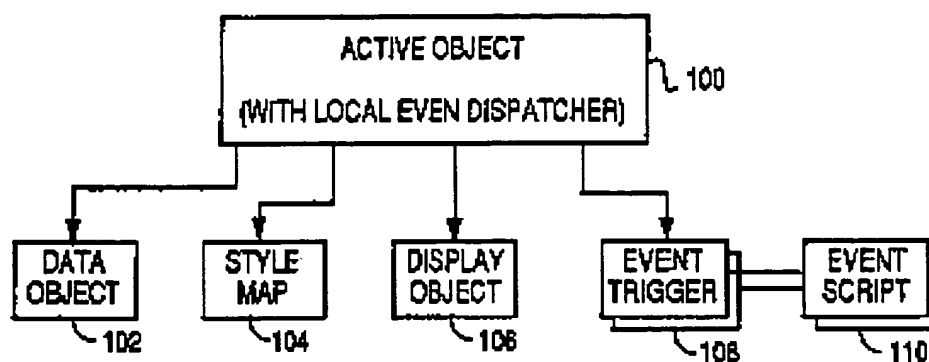


FIG. 7

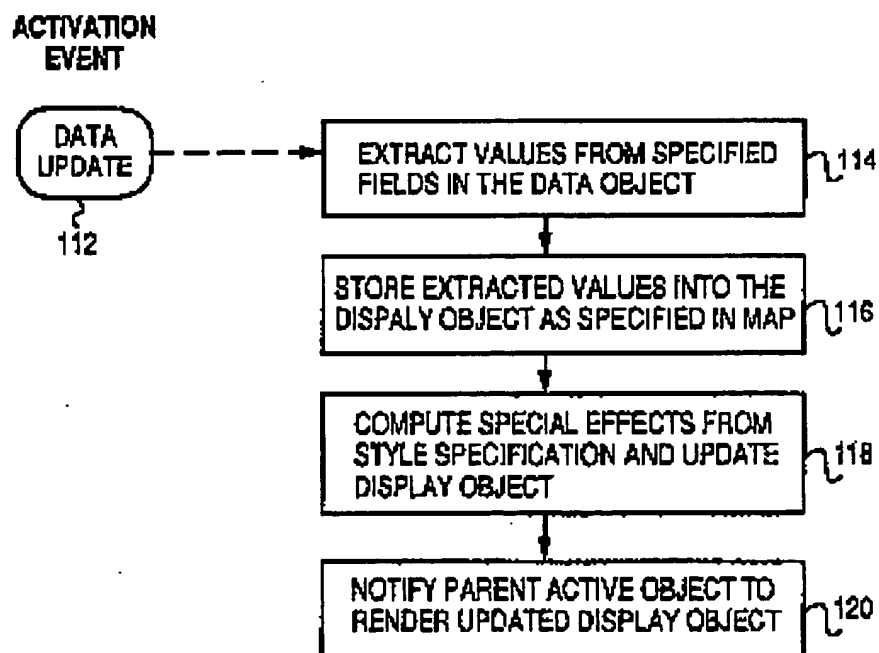


FIG. 8

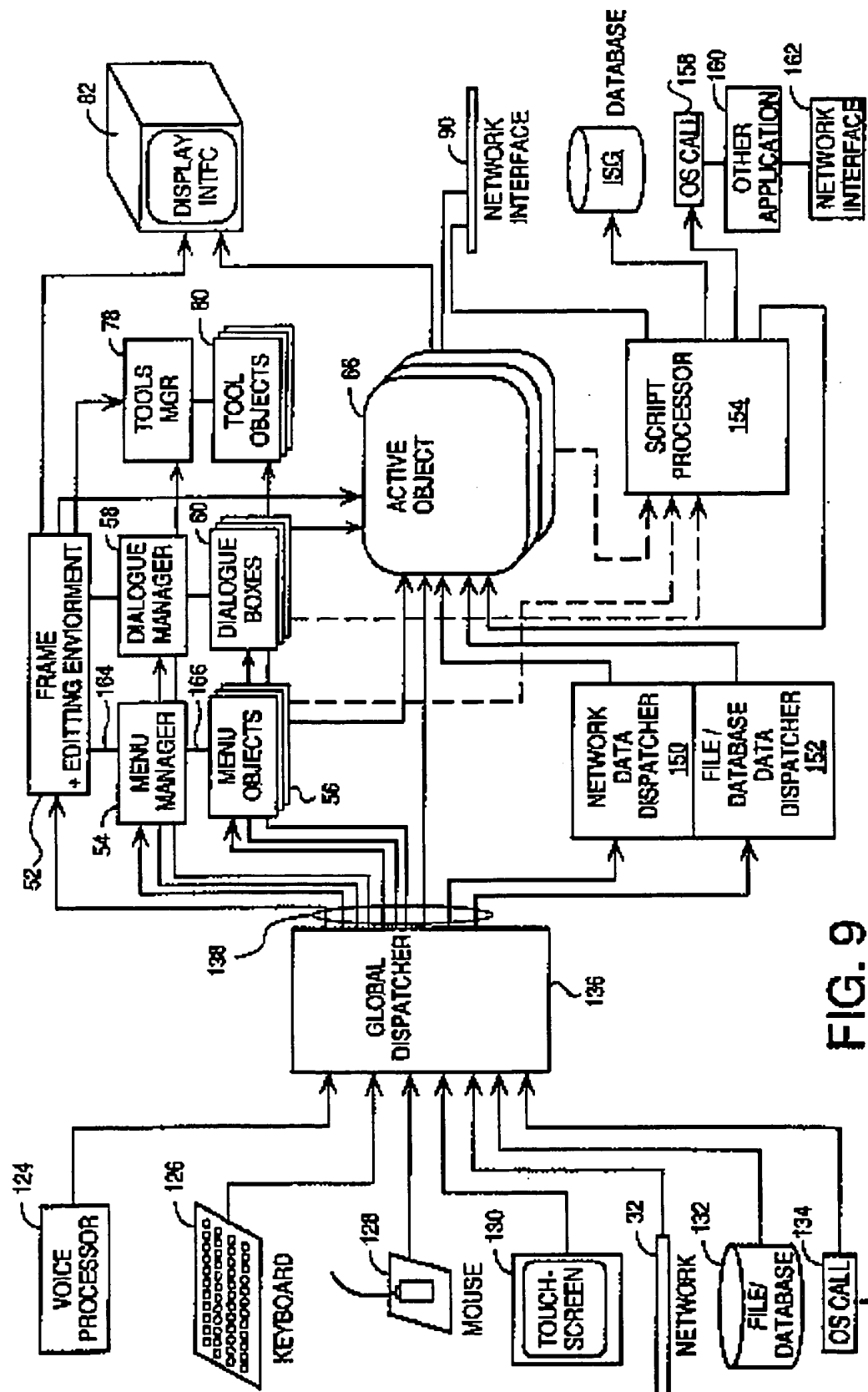


FIG. 9

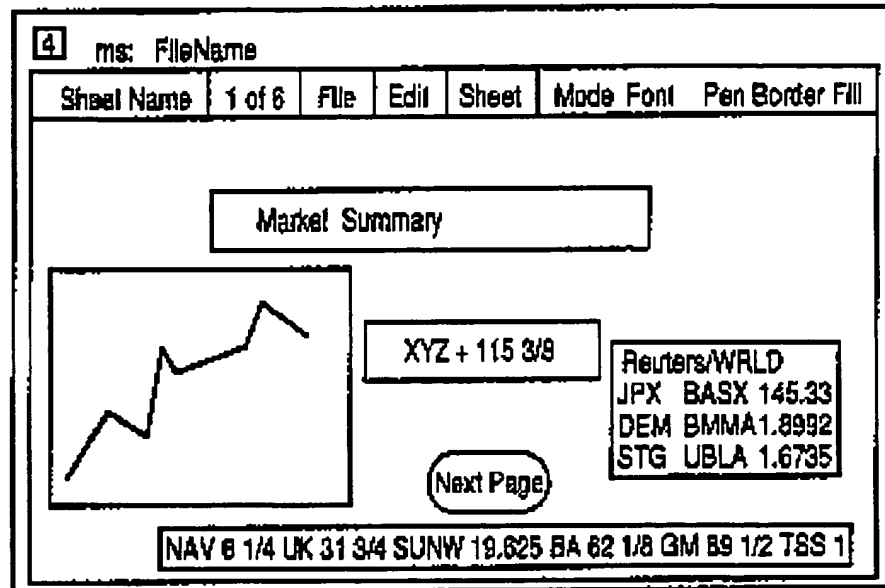


FIG. 10

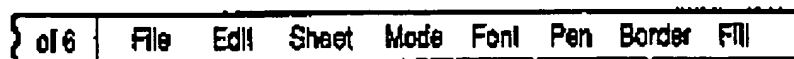


FIG. 11

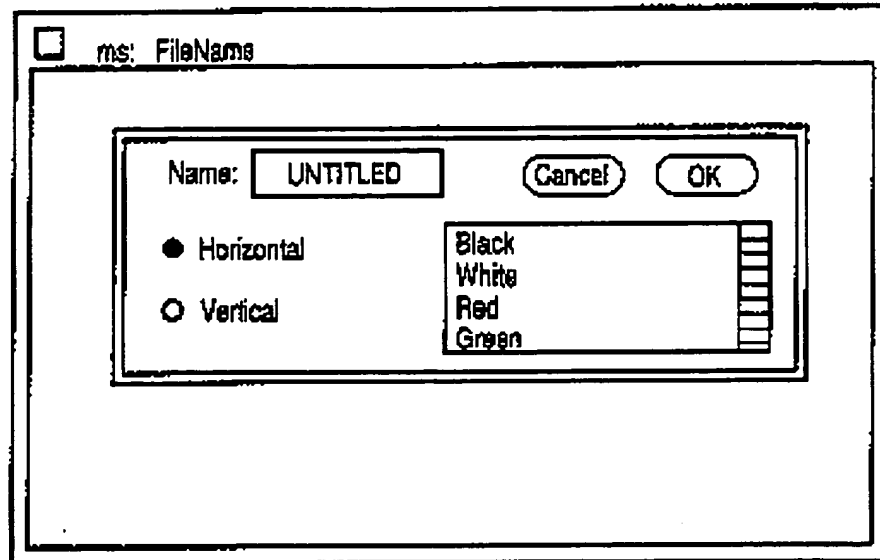


FIG. 12

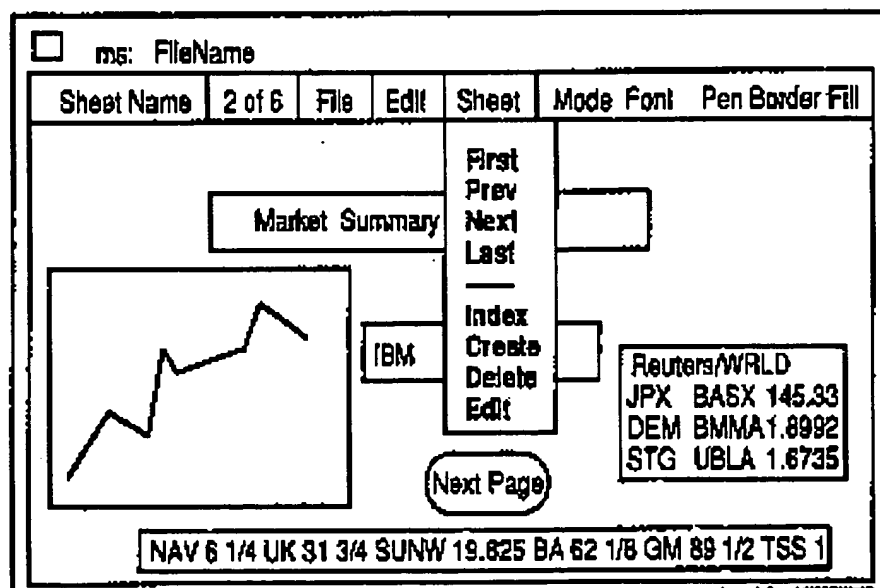


FIG. 13

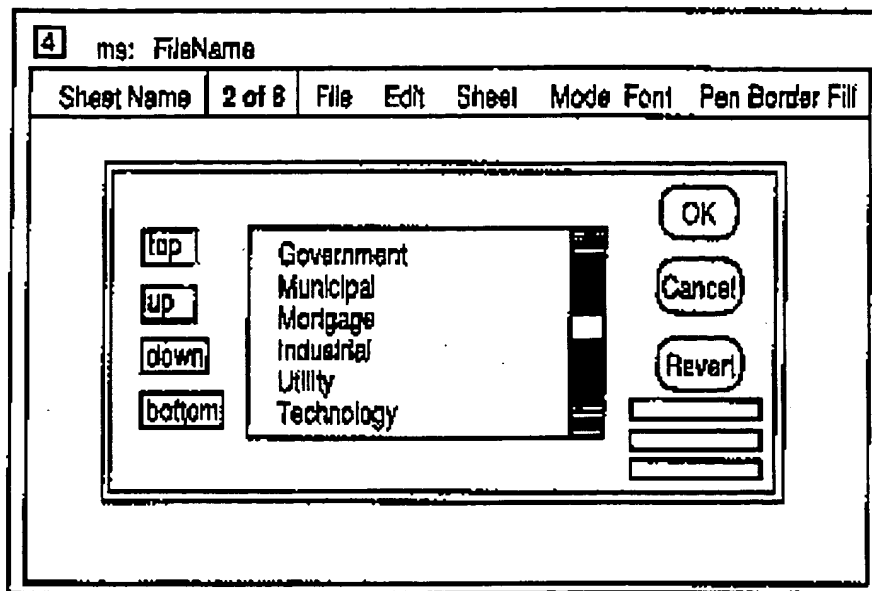


FIG. 14

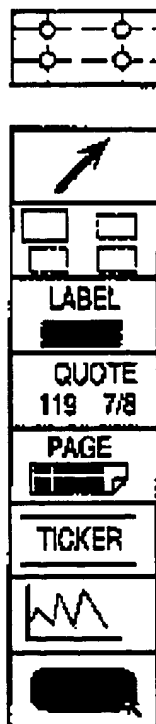


FIG. 15



4/10

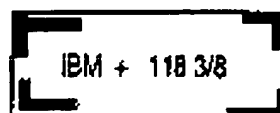


FIG. 16

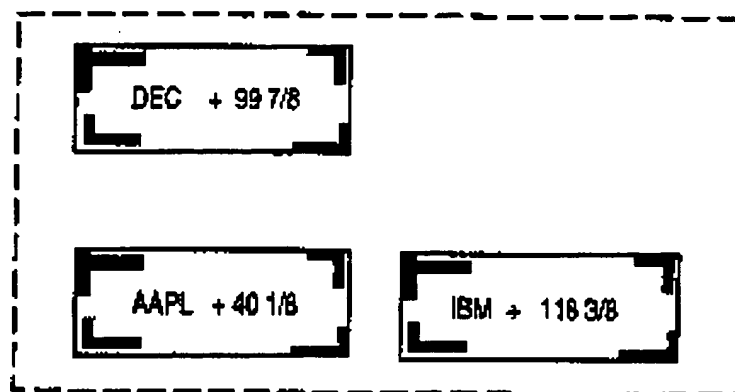


FIG. 17

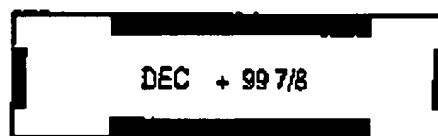


FIG. 18



FIG. 19

SUBSTITUTE SHEET

10/10

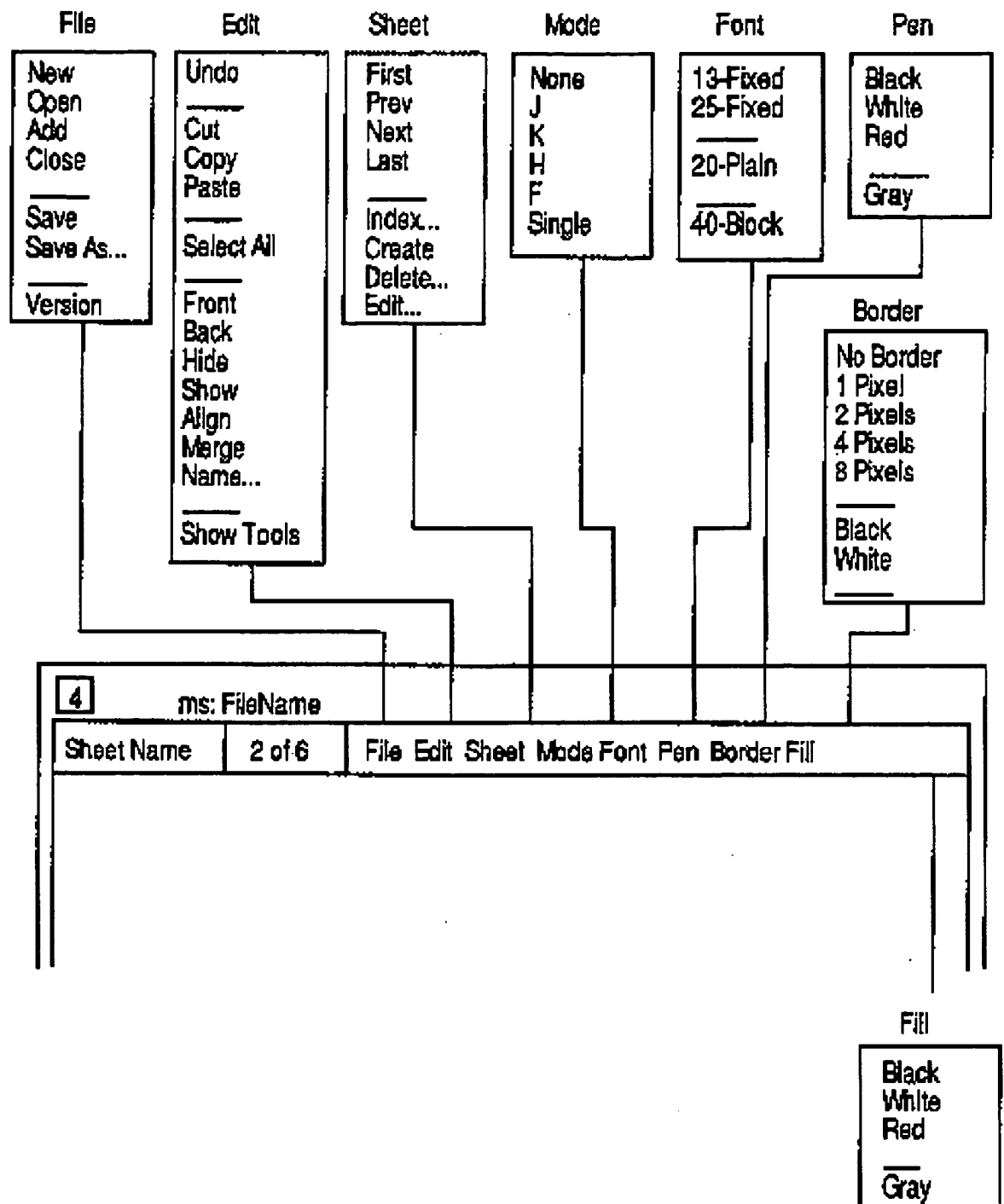


FIG. 20